Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1987

Design of an addressable memory controller.

Ham, Byung Woon.

http://hdl.handle.net/10945/22737

Copyright is reserved by the copyright owner.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DESIGN OF AN ADDRESSABLE
MEMORY CONTROLLER

by

Byung Woon, Ham

March 1987

Thesis Advisor Chin-Hwa, Lee

Approved for public release; distribution is unlimited.

T233104

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 62	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION Naval Ocean Systems Center 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code) San Diego, California 92152	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO PROJECT NO TASK NO WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

DESIGN OF AN ADDRESSABLE MEMORY CONTROLLER

12 PERSONAL AUTHOR(S)

HAM, Byung Woon

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM ______ TO ______	14 DATE OF REPORT (Year, Month, Day) 1987 March	15 PAGE COUNT 83

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Content addressable memory

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The main memory is an essential subsystem in a Von Neumann type of stored program machine. Because of the speed gap existence between the processor and the main memory, there has been a constant need to improve the main memory to achieve a better throughput. One method is to use a CAM(Content Addressable Memory). It is known as a very powerful facility for searching a particular item from a data array rather than from conventional memory. Investigated in this thesis are the discussion of CAM characteristics, timing analysis, CAM controller design and simulation results. The main results obtained in this thesis are timing characteristics of the CAM system and design considerations of the CAM controller

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Chin-Hwa, Lee	22b TELEPHONE (Include Area Code) 408-646-2190	22c OFFICE SYMBOL 62Le

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Approved for public release; distribution is unlimited.

Design of an Addressable Memory Controller

by

Byung Woon. Ham
Lieutenant, Republic of Korean Navy
B.S., Korean Naval Academy, 1979
B.S., Seoul National University, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1987

ABSTRACT

The main memory is an essential subsystem in a Von Neumann type of stored program machine. Because of the speed gap existence between the processor and the main memory, there has been a constant need to improve the main memory to achieve a better throughput. One method is to use a CAM(Content Addressable Memory). It is known as a very powerful facility for searching a particular item from a data array rather than from conventional memory. Investigated in this thesis are the discussion of CAM characteristics, timing analysis, CAM controller design and simulation results. The main results obtained in this thesis are timing characteristics of the CAM system and design considerations of the CAM controller

Thesis
H62175
c.1

TABLE OF CONTENTS

I. INTRODUCTION 10
  A. CONVENTIONAL MEMORY SYSTEM 10
  B. CONTENT ADDRESSABLE MEMORY (CAM) 11
    1. CAM Reading Operation 14
    2. CAM Writing 14
    3. SET Operation 14
    4. SELECT FIRST Operation 14
  C. OUTLINE OF THIS THESIS 16

II. DESIGN OF THE CAM 20
  A. OUTLINE OF THE CAM DESIGN 20
  B. COMPONENT DESIGN 20
    1. FACE Circuit 20
    2. DECODER Circuit 20
    3. CELL Body 23
    4. RESPONDER Circuit 23
  C. THE CAM DESIGN 26

III. CAM PERFORMANCE ANALYSIS 29
  A. CAM TIMING ANALYSIS 29
    1. Write Cycle Timing Analysis 29
    2. Read Cycle Timing Analysis 32
  B. CAM TIMING SIMULATION 36
    1. Four Word CAM Timing 38
    2. Throughput Discussion 41
  C. RESTRICTIONS OF THE CAM SYSTEM 41

IV. CAM CONTROLLER DESIGN 44
  A. DESIGN CONSIDERATIONS FOR CAM CONTROLLER 44

- B. CAM CONTROLLER DESIGN 45
- C. THROUGHPUT DISCUSSION 47

V. CAM TIMING SIMULATION WITH THE CONTROLLER 52
- A. CAM SIMULATION WITH CONTROLLER 52
- B. THROUGHPUT ANALYSIS 56

VI. CONCLUSIONS AND RECOMMENDATIONS 64
- A. CONCLUSIONS 64
- B. RECOMMENDATIONS 64

APPENDIX A: CAM SIMULATION PROGRAM FOR SCALD SYSTEM 66

APPENDIX B: CAM WITH CONTROLLER SIMULATION PROGRAM FOR SCALD SYSTEM 70

APPENDIX C: MICROPROCESSOR(Z-80) TIMING DIAGRAM 77

LIST OF REFERENCES 81

INITIAL DISTRIBUTION LIST 82

LIST OF TABLES

1. SIGNALS AND THEIR ACTIVE VALUES IN THE RESPONDER 26
2. WRITE CYCLE TIMING DIAGRAM SYMBOLS AND TIME 31
3. WRITING SEQUENCE AND THE ACTIVE SIGNAL NAME 32
4. READING SEQUENCE AND THE ACTIVE SIGNALS NAME 34
5. READ CYCLE TIMING DIAGRAM SYMBOLS AND TIME 38
6. FOUR WORD READ CYCLE TIMING VALUES 41
7. CAM READ CYCLE TIME BASED ON SIZE 42
8. TRANSITION TABLE OF CONTROLLER 47
9. TRUTH TABLE FOR SIGNAL DEN 51
10. RAM WRITE PROGRAM FOR Z80 59
11. CAM WRITE PROGRAM FOR Z80 60
12. CAM AND CONVENTIONAL MEMORY WRITE TIMING 60
13. RAM READING PROGRAM Z80 61
14. CAM READING PROGRAM FOR Z80 62
15. CAM AND CONVENTIONAL MEMORY READ TIMING 63

LIST OF FIGURES

1.1 Conventional Memory System 11
1.2 Overviews of CAM 12
1.3 Masked Data Generation Circuitry 13
1.4 The Memory Array of a CAM 15
1.5 CAM Read Operation 16
1.6 Write Circuits for CAM 17
1.7 SELECT FIRST Operation in the RESPONDER 18
2.1 FACE Circuit 21
2.2 FACE Body 21
2.3 DECODER Body 22
2.4 DECODER Circuit 22
2.5 CELL Body 23
2.6 CELL Circuit 24
2.7 RESPONDER Body 24
2.8 RESPONDER Circuit 25
2.9 CAM System Body 27
2.10 CAM System Diagram 28
3.1 FACE Timing Diagram in the Write Cycle 29
3.2 CELL Timing Diagram in the Write Cycle 30
3.3 RESPONDER Timing Diagram in the Write Cycle 30
3.4 DECODER Timing Diagram in the Write Cycle 31
3.5 CAM Write Cycle Timing Diagram 33
3.6 FACE Timing Diagram in the Read Cycle 34
3.7 CELL Timing Diagram in the Read Cycle 35
3.8 RESPONDER Timing Diagram in the Read Cycle 36
3.9 CAM Timing Diagram in the Read Cycle 37
3.10 CAM Simulation Waveform Diagram 39
3.11 CAM Simulation Waveform Diagram (Continued) 39

4.1 CAM Controller Body 46
4.2 CAM Controller Circuit 48
4.3 Controller Write Cycle Timing Diagram 48
4.4 controller Read Cycle Timing Diagram 49
5.1 Simulation Circuit Diagram 53
5.2 CAM with Controller Simulation Waveform Design 53
5.3 CAM with Controller Simulation Waveform Diagram (Continued) 54
5.4 Comparison of CAM and Conventional Memory Write Timing 57
5.5 Comparison of CAM and Conventional Memory Read Timing 58
C.1 WAIT Request Timing Diagram 77
C.2 Memory Read Cycle Timing Diagram 78
C.3 Memory Write Cycle Timing Diagram 78
C.4 Output Timing Diagram 79

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to Professor Chin-Hwa Lee of the Department of Electrical and Computer Engineering of the Naval Postgraduate School for the Guidance, assistance and continuous encouragement in the pursuit of this project. Also, thanks to Professor Sherif Michael for his encouragement.

I thank my wife Byung Ok and my son Jung Woo for their support while I was away from our home during my stay in the United States, and my mother for her love. I dedicate this thesis to the gravestone of my sister Jin Sook.

I. INTRODUCTION

In digital systems, the main memory subsystem is essential in a Von Neumann type of stored program machines. This is the key feature of modern computers which allows the instructions to be held the main memory while awaiting execution. The instructions have to be fetched to the CPU by assigning proper address signals to the system bus.

During the past three decades, computer systems have grown from simple conventional Von Neumann machines to complicated virtual time sharing systems. This growth has brought profound changes in the organization of both the processor and the memory system. During this period, memory systems have had an improvement of about 1000 times in speed and 500 times in capacity while the processing power of the computer has increased by a factor of 10 every five years. [Ref. 1]. In spite of these improvements for both the processors and the main memory systems, there has been a persistent and severe mismatch between the speed of the processor and that of the main memory. Hence, there has been a constant need to improve main memory to achieve better throughput.

A. CONVENTIONAL MEMORY SYSTEM

The main memory in a usual computer system has both the address register and the data buffer register through which information is passed to the CPU. The functional organization of the main memory is shown in Figure 1.1. The memory consists of an array of cells for storing the information, an address register, and a data buffer register. The data and address busses are controlled by a memory controller. In small computer systems the controller may just initiate a read or write operation. In more sophisticated systems, the memory control logic permits all processors in the system to operate at full speed with minimal interference. The address bus is unidirectional while the data bus is bidirectional. Memory must receive an address but may either receive data (for a write operation), or transmit data (for a read operation). To read information out of a conventional memory, the system provides the address of the cell whose contents are wanted.

A memory read operation similar to instruction fetch is done in sequential manner. This sequential read operation may be a serious disadvantage when a

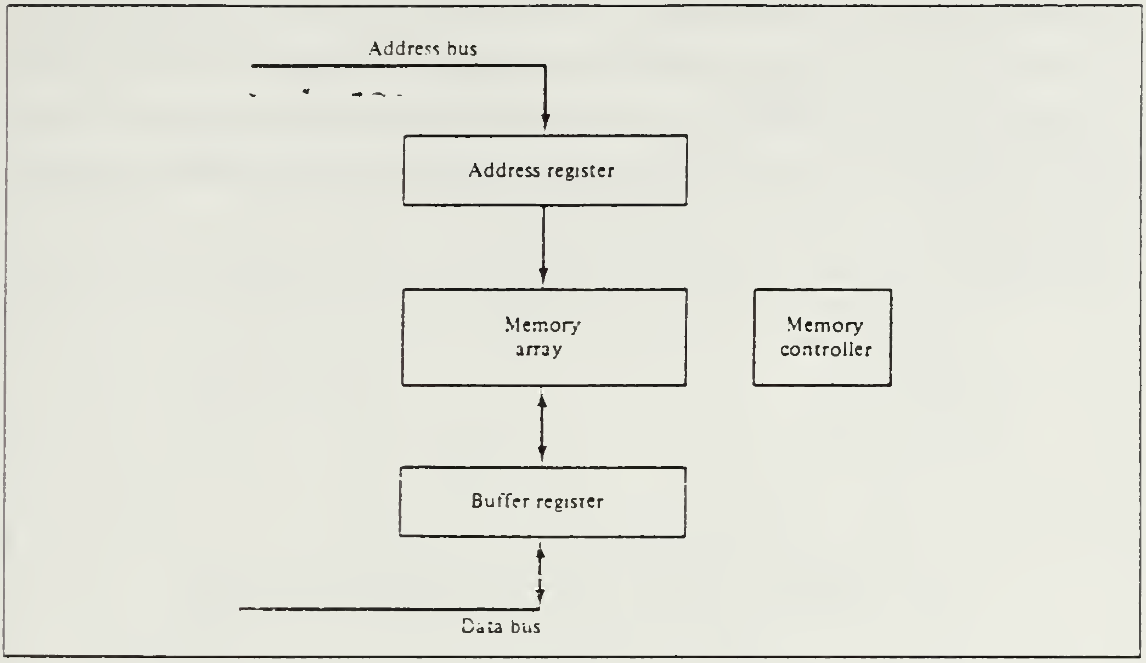


Figure 1.1 Conventional Memory System.

particular item is needed from the big data array. It takes too much time to search the object one by one sequentially in location even though many improved methods have been developed. When one item is needed from the data array, the total time spent for searching depends on the size of the memory in the conventional memory system. If the size of the array is N, the average number of operation for searching is $(N+1)/2$ using a simple linear searching method or Log(N) using a binary searching method [Ref. 2].

B. CONTENT ADDRESSABLE MEMORY (CAM)

Progress in the computer industry generally has been guided by the principle of "better performance with minimum cost" that has resulted in a variety of innovative techniques for improving the performance of the computer system [Ref. 1]. These innovation can essentially be divided into following two categories.

- Improving the performance by increasing component speed.
- Improving the performance by architectural innovations.

However, the speed of the components is limited by technology. Hence the first method loses its appeal eventually, and it necessitates increase of the computational throughput by some other means. One of these methods is using CAM (Content

Addressable Memory), sometimes called a "Distributed Logic Machine" [Ref. 3]. When a program needs to search for a specific item in the data array, the masked target data is broadcast to the CAM array in parallel. Each CAM cell compares this masked data with its contents. If its contents are the same as the masked data, the tag bit is set and the corresponding position is shown in the data gathering device shown in Figure 1.2 after some delay.

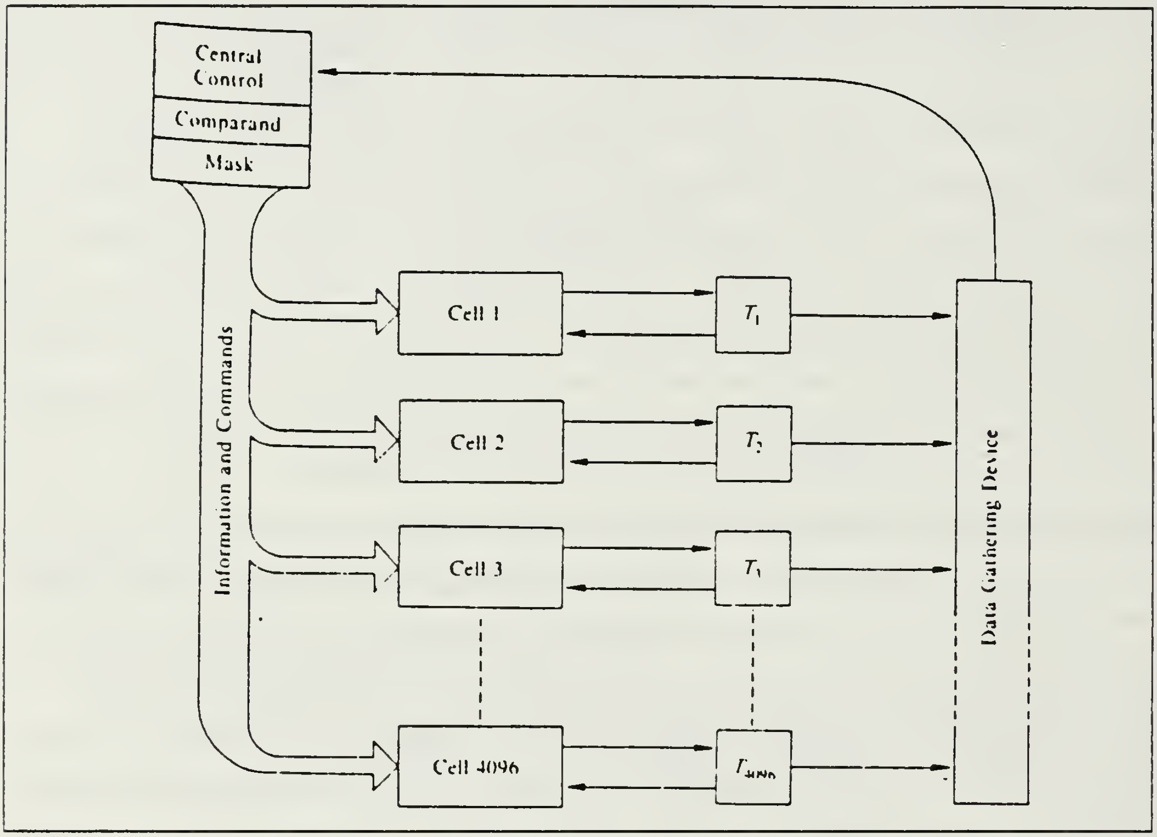


Figure 1.2 Overviews of CAM.

Since the data is broadcast to all cells, only one time memory access is needed in the CAM. The number of memory read operations of the CAM for the search is always one while that of the conventional memory depends on the size of memory. This parallel read operation is a unique characteristic of the CAM system.

The comparand register shown in Figure 1.3 is used to hold the data to be compared in the CAM. The mask register allows partial matching of the data. When a portion of a memory word is needed, corresponding bits of the mask register are set to high while the other bits are set to low. The comparand register, modified by the masked bits, are broadcast to the CAM array. From Figure 1.3, if the jth bit of the mask is zero, neither match line $M1_j$ nor MZ_j will be activated. If mask contains 1 in this position, then depending on whether the comparand has 0 or 1 stored there, one of the match lines will be asserted and the other will be left inactive. These two match lines, $M1_j$ and MZ_j, go to the jth bit of every CAM cell. The above relationship can be expressed in the following question.

$$(M1_j \bullet S_{ij}) + (MZ_j \bullet S_{ij})$$

where S_{ij} stands for the jth bit of ith cell.

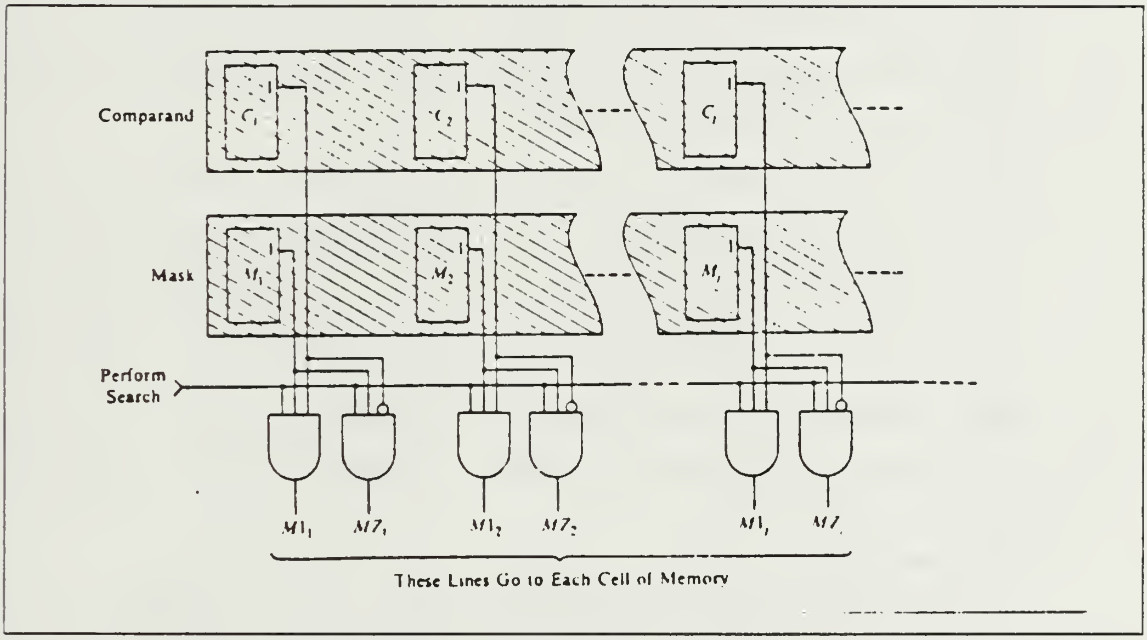


Figure 1.3 Masked Data Generation Circuitry.

If the system looking for an one and S_{ij} is zero, then a mismatch signal will be generated at this bit position and passed down the mismatch signal line through the next row of cells to reset the tag bit of this cell. If a word stored in this cell disagrees with the comparand at more than one bit position for which the mask contains ones, the effect is the same as if only one bit mismatch occured. If a word agrees with the contents of the comparand register only in a place or places where the mask register contains zeros then no mismatch signal is generated and the tag bit of that word remains set.

1. CAM Reading Operation

To select a particular word for reading in a conventional memory, we need to present the address to the system. But in a CAM there is no address associated with a cell. All we have to distinguish one cell from another is whether or not they are responders, i.e., have their tag bits set on. We will therefore make a virtue out of necessity and arrange to read out the contents of any cell that is a responder. If more than one word is a responder the read lines (R_i in Figure 1.4) will contain the logical OR of the contents of all responders. The diagram of CAM reading is shown in Figure 1.5.

2. CAM Writing

The basic CAM writing operation is Multiwrite. This is an ability to write, in parallel, into as many words as pointed by the responders, all at the same time. Just as we used the responder/nonresponder distinction to reset the cell to be read from, the same criterion can be used to decide which words are to be written into. Figure 1.6 shows the write circuits at each bit of memory.

If we wish to write a one into the jth bit of all responders we energize the $W1_j$ line. Where the responder lines are energized, a one will be stored. Where the responder line is not energized, no change in the information stored will take place. If we don't wish to write into some particular bit of the responders we simply don't energize either the W1 or the WZ line of that bit. Another method to write data into the CAM is to use a decoder. The function of the decoder is to set the responder's tag bit before writing a particular datum into the CAM. Using a decoder, data can be written into any position of the CAM system.

3. SET Operation

Setting the tag bits of the responders is needed in another read cycle since the all tag bits were already cleared in the previous read cycle by the SELECT FIRST operation. This requires some circuitry. Figure 1.7 shows a line called the SET line which runs to the set input of each and every tag bit. When energized by the controller, it will turn on all the tag bits.

4. SELECT FIRST Operation

It is possible that more than one cell of the memory responds to some search. Further, there will be occasions when we wish to single out just one of these responders and deal with it alone. To provide this capability, we organize the cells of the memory to the extent that each cell has a predecessor and a successor. In this

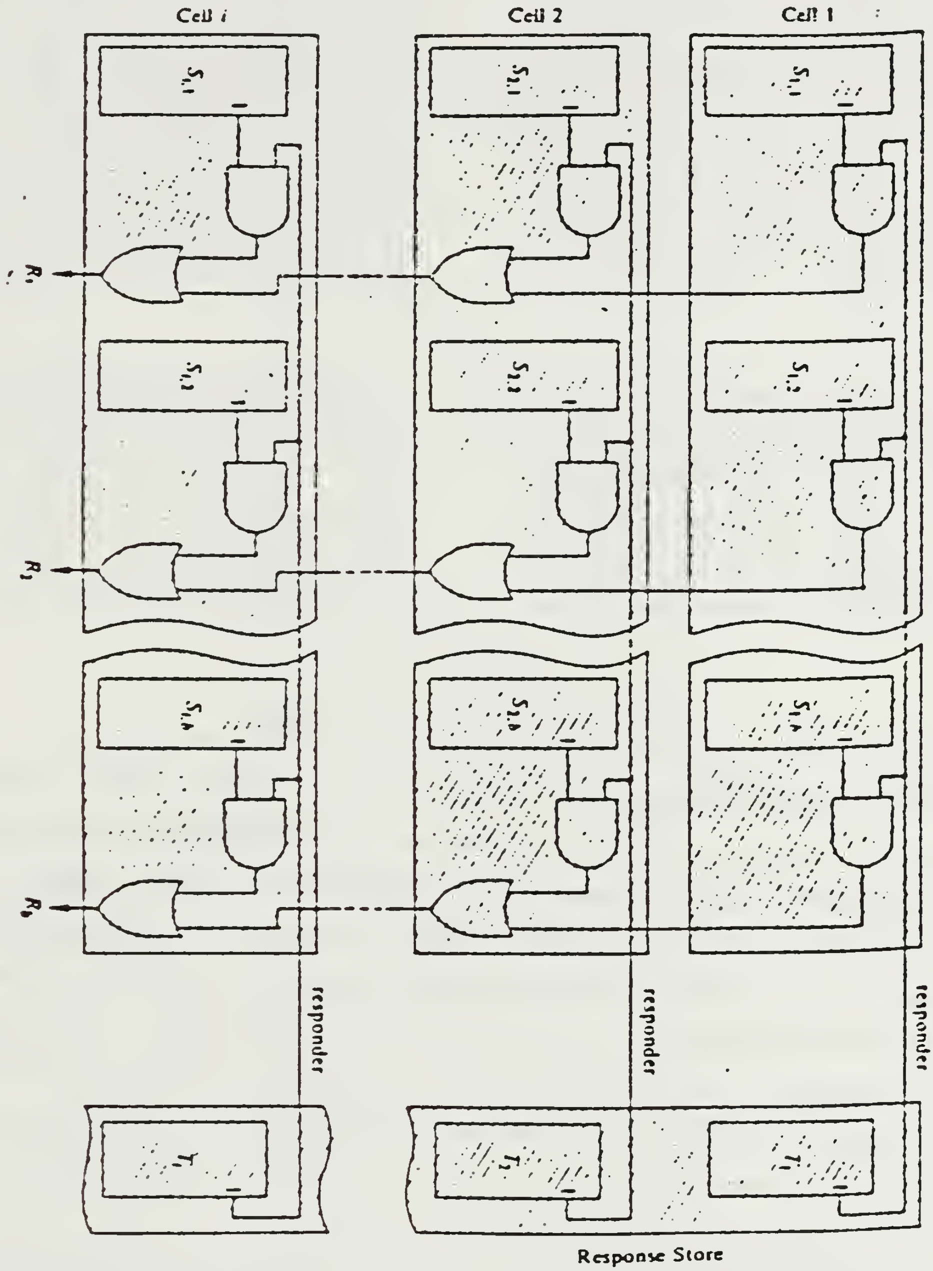


Figure 1.4 The Memory Array of a CAM.

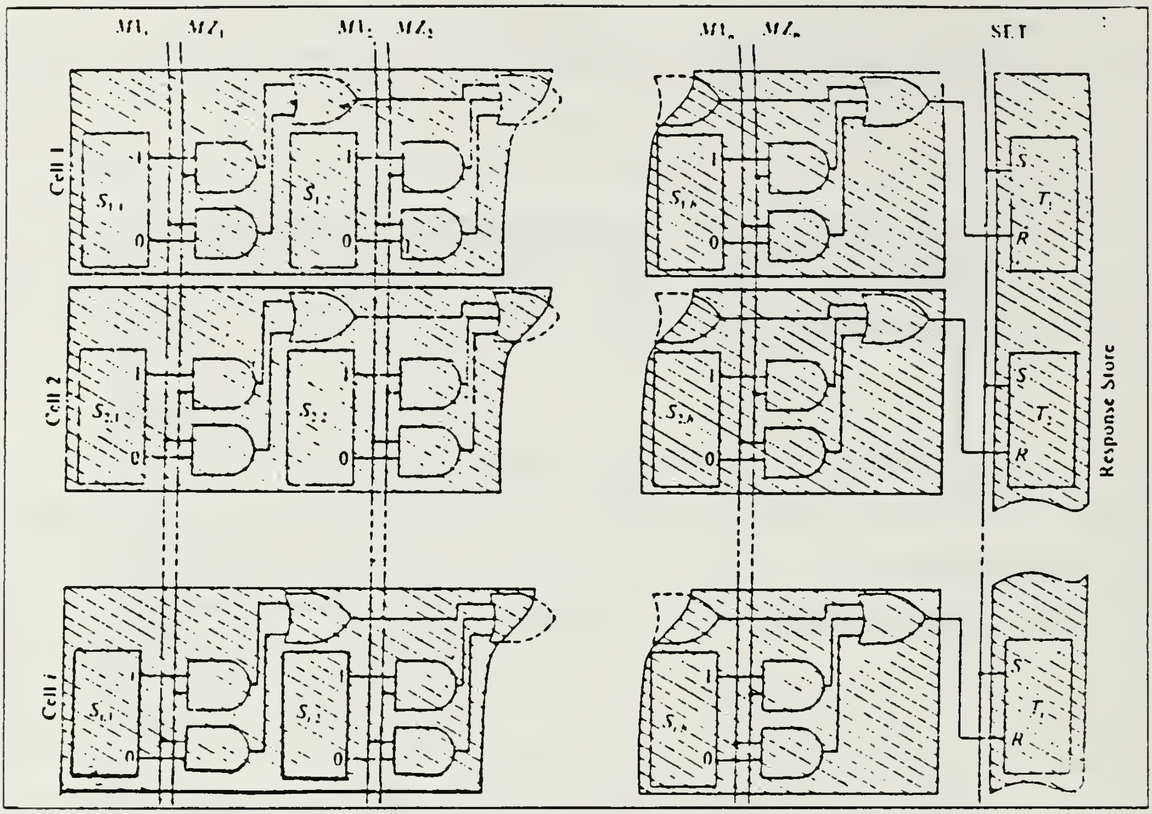


Figure 1.5 CAM Read Operation.

sense, there is always an "earliest" or "first"responder, the closest one to the top of the CAM array. Using the circuit of Figure 1.7, the SELECT FIRST operation is supported. This turns off the tag bit of any cell whose predecessor, the predecessor of whose predecessor, or indeed any of whose "ancestors" is on. Then the SOME/NONE line at the point labelled "a" will have no signal on it because no earlier cell has a tag bit turned on. But at point "b" and at all following cells the SOME/NONE line will be energized by the fact that $T_j = 1$. Then when the select first line is energized, each of these succeeding tag bits will be reset just as if they had held a mismatch. The SOME NONE line is also used to tell the controller that some, or none, of the cells have their tag bits set.

C. OUTLINE OF THIS THESIS

In this thesis, the SCALD system is used as a CAD tool. All circuits and their timing diagrams are extracted from the results of the SCALD system. The analysis of the required circuits are split into small blocks depending on their function. They are

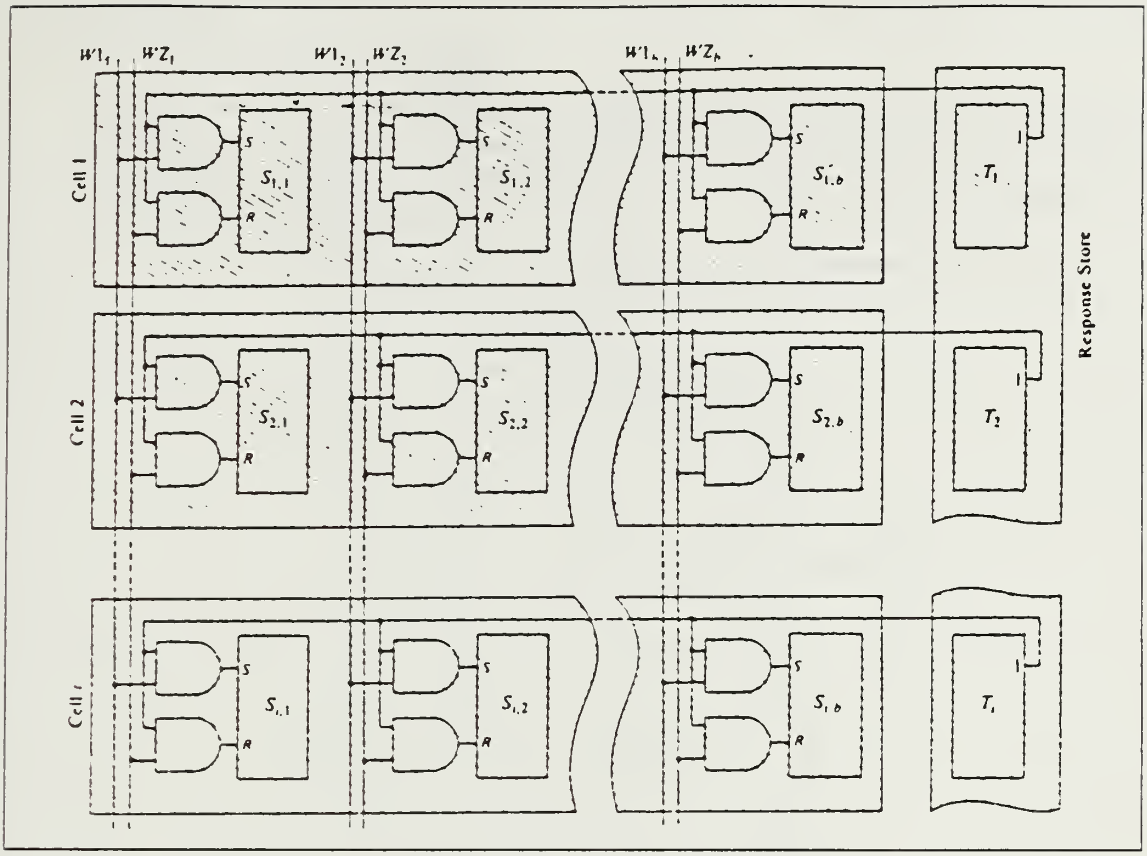


Figure 1.6 Write Circuits for CAM.

combined together to build a CAM system. This is possible because the SCALD system provides a hierarchical design environment.

In Chapter II, each block is built and combined to make the total CAM. The timing characteristics of each block will be discussed and those of the CAM are considered in Chapter III. Also, the CAM simulation will be done with results of the timing analysis. In Chapter IV, the CAM controller design is discussed because of the difference between the CAM read and the write cycle time. Using timings of the standard bus in the PROLOG system with a Z-80 microprocessor and four word CAM, the simulation is done in Chapter V. This is a timing adaptation of the CAM system to the CPU. Finally, a conclusion will be presented. This thesis also contains appendices which provide the listing of the CAM simulation programs for the SCALD system and the timing diagrams of the PROLOG system. The main results obtained include:

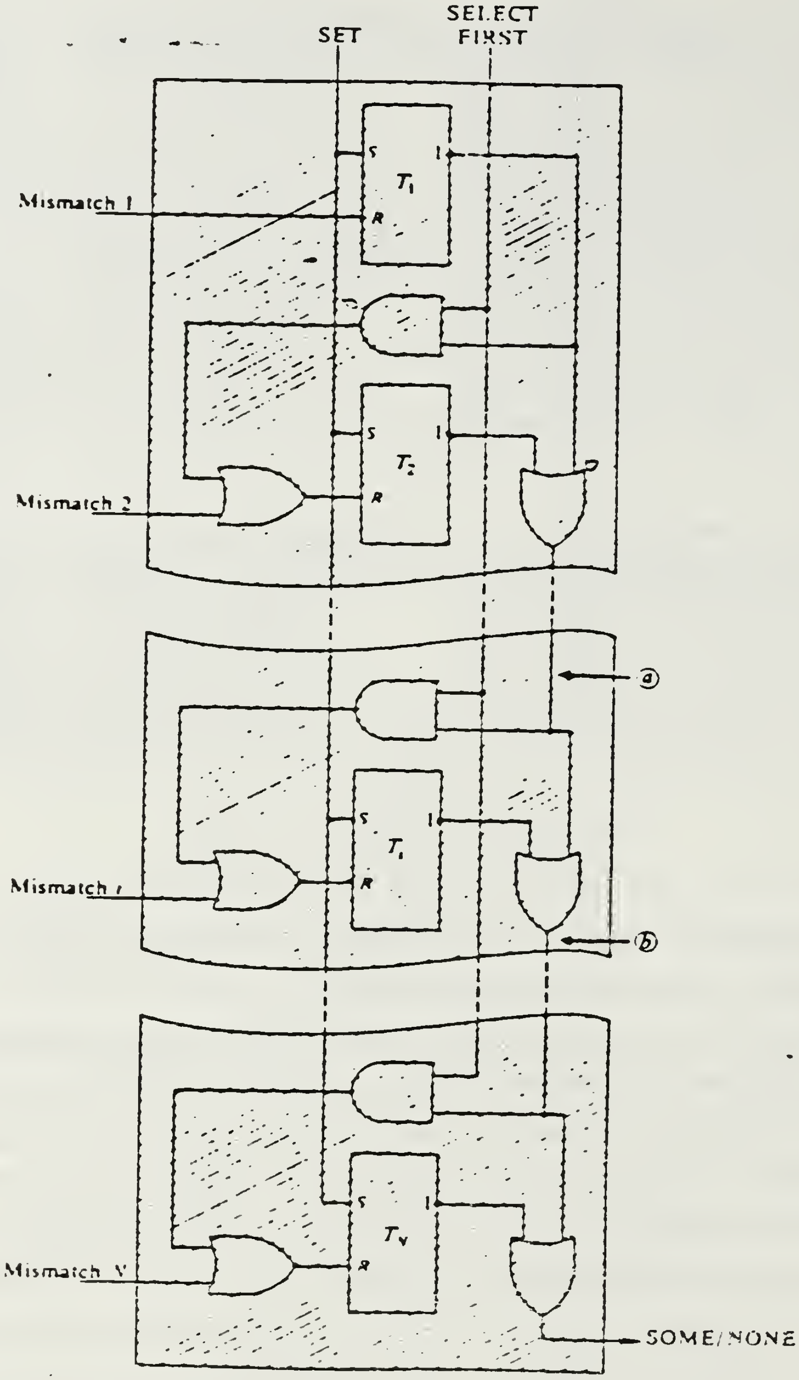


Figure 1.7 SELECT FIRST Operation in the RESPONDER.

- Design of a simple CAM system with a CAD tool by using a hierarchical design concepts.
- CAM Read/Write cycle timings.
- CAM Controller design consideration based on the results of the CAM timing analysis.
- Throughput analysis of the CAM and the Conventional memory system in both the read the write cycles.

- II. DESIGN OF THE CAM

A. OUTLINE OF THE CAM DESIGN

The SCALD system allows hierarchical designs. If one portion of the logic circuit can be separated out as a block, it can be used repeatedly by calling that part in other circuit designs. The CAM circuit in this thesis is built using this hierarchical concept. There are four major blocks in the CAM. Their names and functions are

- CELL - - individual memory cell including matching circuitry.
- FACE - - generates masked signals by receiving WR, RD, mask and comparand. It supplies them to all column CELLs.
- DECODER - - address decoder.
- RESPONDER - - brain of the CAM system. Contains tag bit to do read/write operation by receiving SELECT FIRST, SET, MISMATCH, and SELECT signals. Also the RESPONDER does the SELECT FIRST operation during the read cycle.

The number of CELLs are dependent on the total CAM size and can be calculated by multiplying the word length (N) with the number of words (M). The total number of FACEs are the same as word length (N) while that of the RESPONDER are a linear function of the number of word (M). Only one DECODER is needed in the whole CAM circuit. The overall diagram of the CAM is shown in Figure 2.10.

B. COMPONENT DESIGN

1. FACE Circuit

As shows in Figure 2.1 the FACE circuit produces masked data bits by receiving comparand and mask bits as operands and RD or WR as the enable signal. The outputs of FACE, W1 and W0 for write cycle, and M0 and M1 for read cycle, are passed to all column CELLs.

2. DECODER Circuit

The basic function of the DECODER circuit is address decoding. The inputs of the DECODER are addresses and the enable signal (DEN). It generates proper decoded signals and passes them to the RESPONDER while the enable signal is high. The Inverter which is connected to the output pins of the LS138 shown in Figure 2.4 change the active low signals to high the RESPONDER requires high when active.

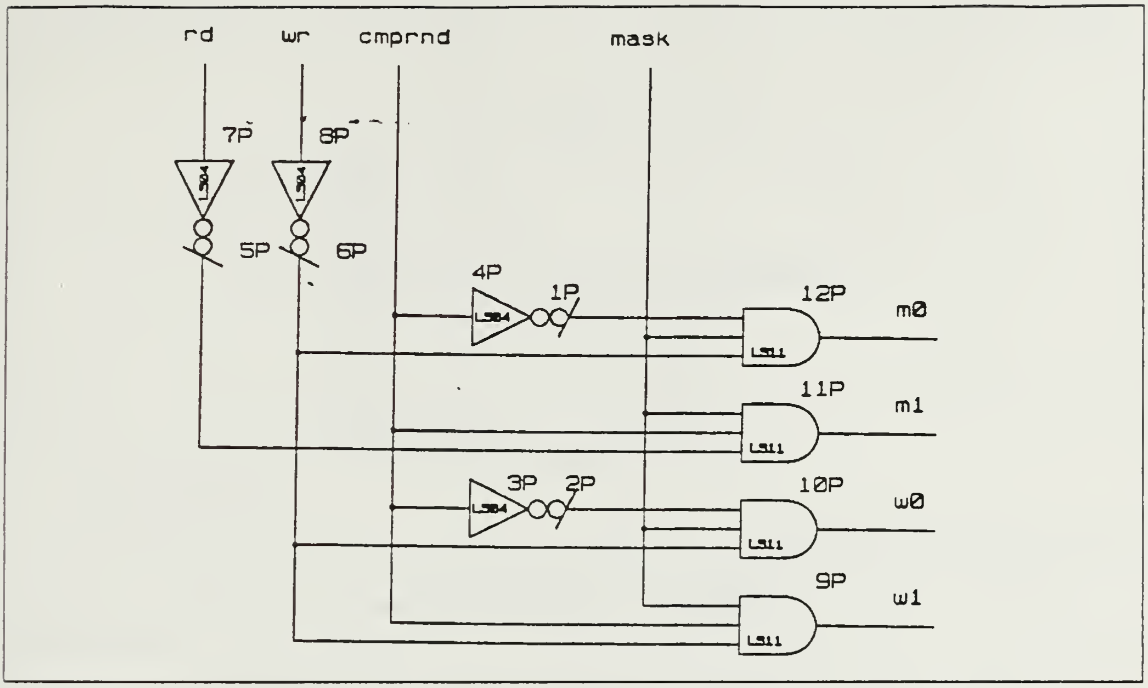


Figure 2.1 FACE Circuit.

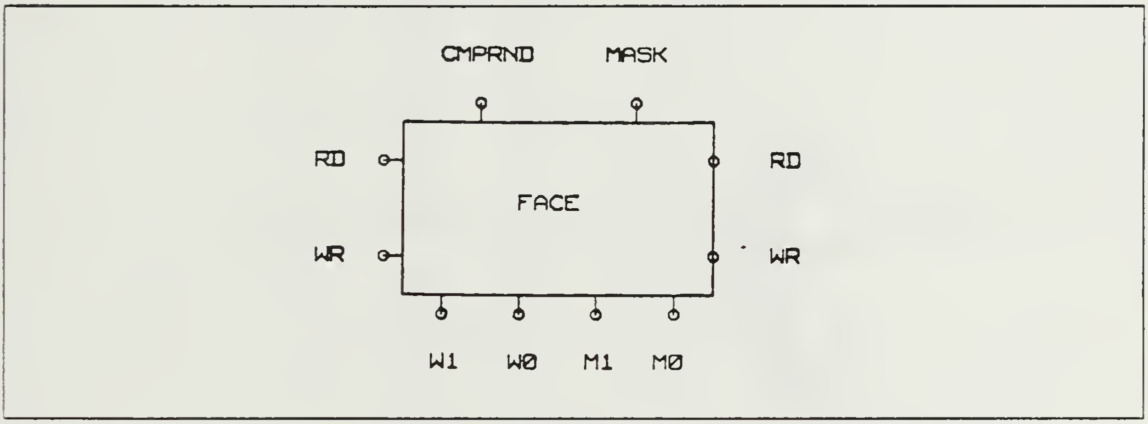


Figure 2.2 FACE Body.

This DECODER can handle a three address bus only, but it can be expanded by cascading and parallel connection. The DECODER need not be activated during the read cycle because the CAM structures does not need decoded address signals.

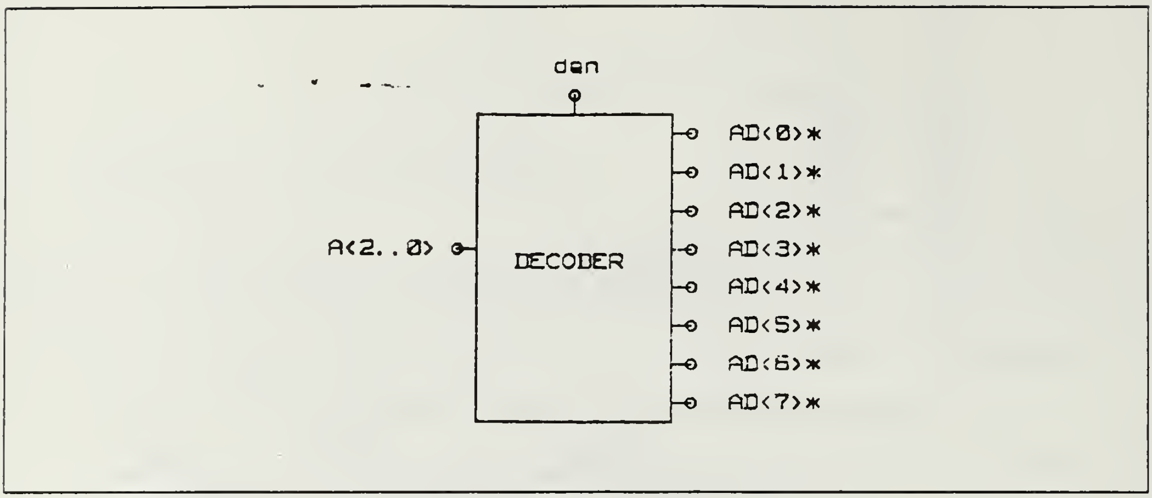


Figure 2.3 DECODER Body.

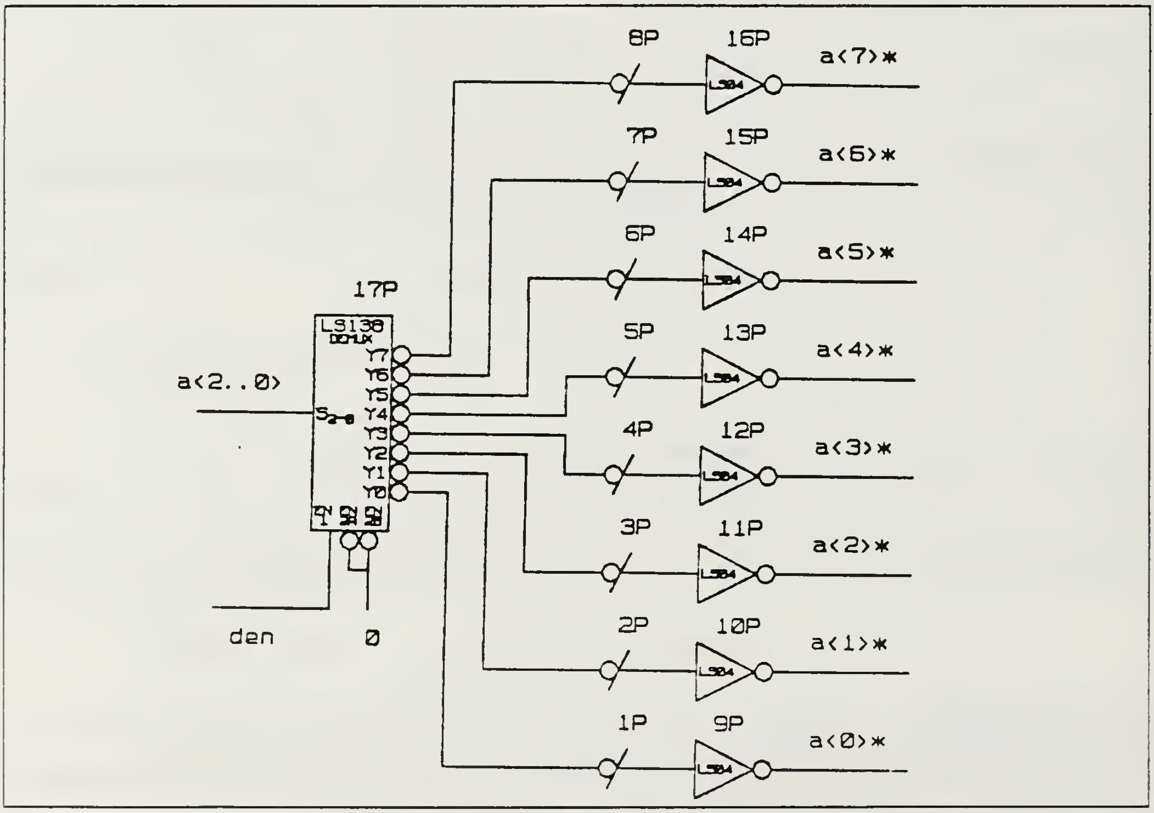


Figure 2.4 DECODER Circuit.

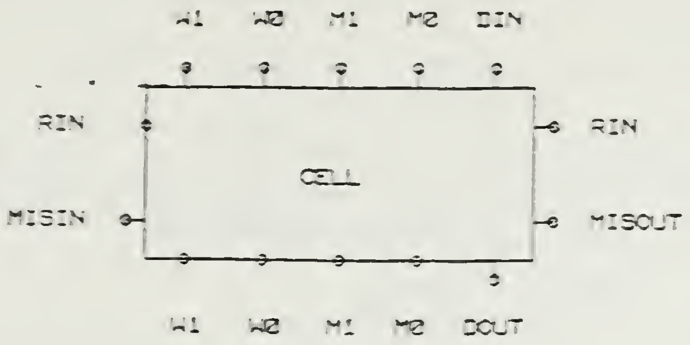


Figure 2.5 CELL Body.

3. CELL Body

The basic functions of the CELL circuit are information store and matching operations. During the write cycle, the RIN, which is the same as RO in the RESPONDER, may be high if we want to write data into this row cells. With W1 and W0 which are the outputs of the FACE, the data is stored in the CELL circuit.

In the read cycle, the high state of RIN is transfered from the RESPONDER which was set to high by the SET signal. But, the high RIN is meaningless at this time because the high SET turns on all tag bits in the RESPONDER. The information which is shown on the data bus may be garbage because the contents of all CELLs appear on the data bus through an OR gate (3P). M1 and M0 are used to decide the status of MISMATCH line with the previous row MISMATCH outputs. The status of the MISMATCH is transfered to the RESPONDER via the next row CELLs. By deciding the proper R0 signal in the RESPONDER, which is denoted as RIN in Figure 2.6, the data line D0 holds the desired data.

If the data desired to read and masked in the FACE with the comparand and mask bits did not match with the contents of the cell, only zeros appear on the data bus. It is because the high MISMATCH clears all RESPONDERs' tag bits and no active R0 is transmitted to the CELLs.

4. RESPONDER Circuit

The RESPONDER is the most complicated part of the CAM system. The functions of the RESPONDER are :

- To generate a read or write enable signal which is designated as RO by the SET, SELECT, MISMATCH and SELECT FIRST with QIN.
- To perform the tag bit clear operation if the SF and the QIN, which are the outputs of the previous stage RESPONDER, are high.

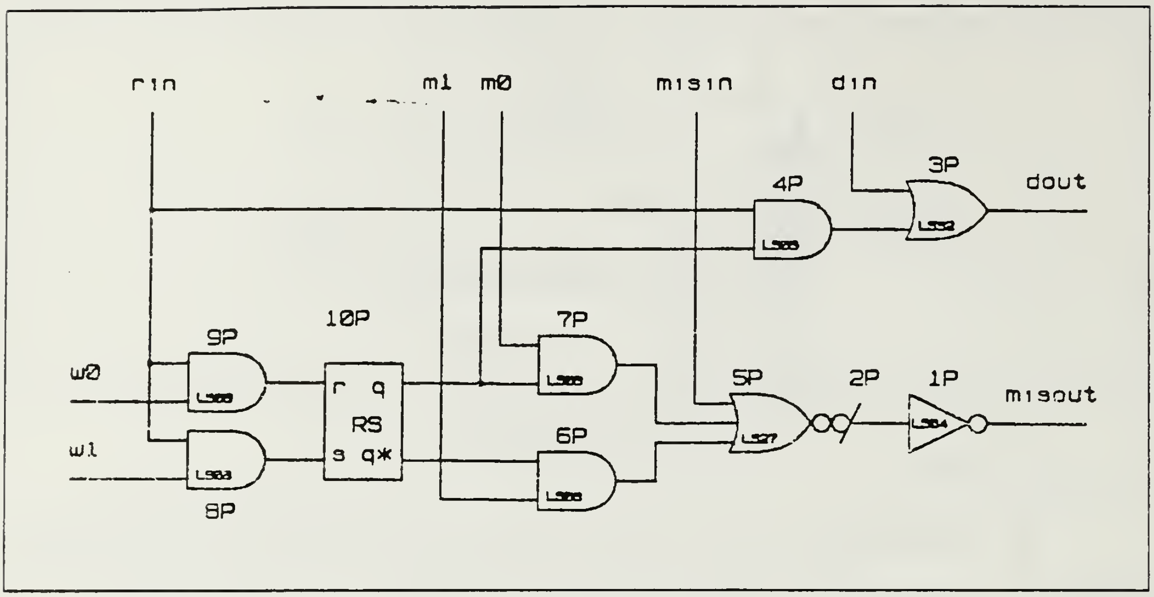


Figure 2.6 CELL Circuit.

From Figure 2.8, the SELECT is an active high input signal which comes from the DECODER during the write cycle only. When the SELECT is high, RS Latch 1 (9P) is set to high and RS Latch 2 (8P) is reset to low. This operation is needed to avoid the undefined state which may cause unexpected result in the system.

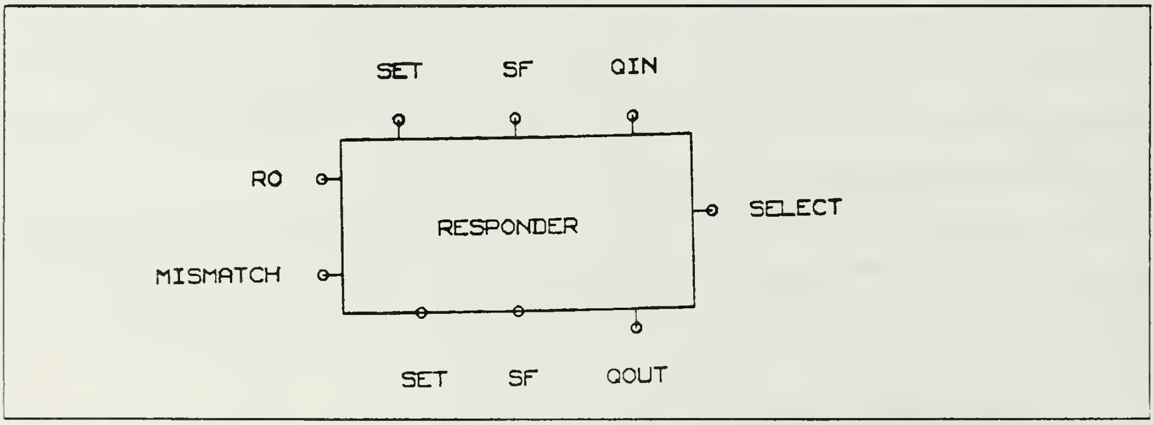


Figure 2.7 RESPONDER Body.

The high R0 is transfered to the column CELLs to do the write operation. When SELECT comes to low, RS Latch 1 becomes low, and RS Latch 2 stays low, and R0 becomes low as a result. If there is no circuitry to reset the RS Latch 1, the R0 signal

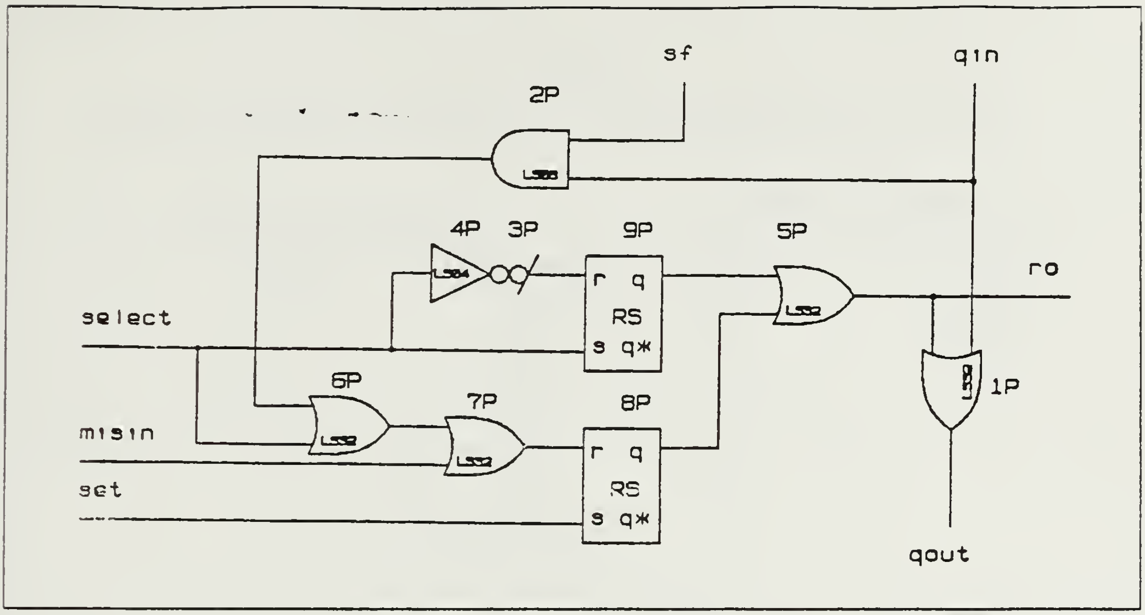


Figure 2.8 RESPONDER Circuit.

stays high until the other signal is activated to reset this Latch. This high R0 may cause errors in another write cycle because if W0 and W1 are already high, this undesired high R0 can change the contents of the row CELLs by overwriting the information. So the RS Latch reset operation is needed after the SELECT returns to low.

In the read cycle, the SET signal first goes to high to set the tag bits in the RESPONDER and generate the high R0 signal. When the SET returns back to low, the RS Latch 2 stays high until the MISMATCH or the combination of SF and QIN are high. The MISMATCH signal may go high if both M0 and M1 are unmatched with the contents of the CELLs or low if matched. The RS Latch is cleared if the MISMATCH is high or remains high if the MISMATCH is low. At this time, the SF signal goes to high to perform the SELECT FIRST operation after the last stage QOUT is available through the OR gates. The SELECT FIRST operation produces only one high R0 signal in the RESPONDERs if many high tag bits exist and this high R0 is transmitted to the RIN of the CELLs to read the desired information. If all row CELLs' MISMATCH signals are high, all tag bits are cleared and only zeros appear on the data bus.

TABLE 1

SIGNALS AND THEIR ACTIVE VALUES IN THE RESPONDER

WRITING CYCLE		READING CYCLE	
Signal Name	Default	Signal Name	Default
RO	H	RO	H
SELECT	H	QIN	H
		QUIT	H
		SF	H
		SET	H
		MISMATCH	H

C. THE CAM DESIGN

Because all blocks were already designed in the previous sections, the CAM system can be built by using these DECODER, CELL, FACE and RESPONDER blocks. All those block parts have their own bodies and each body represents its own circuits. All pin names of the body were carefully assigned for the connection of other bodies, including itself. In this thesis, four word CAM, whose word length is one byte, was built. This four word CAM system will be used for performance analysis and timing simulation in the following Chapters. The total number of the CELLs is 32, which is a product of 4 and 8. The number of the FACEs is 8 while that of the RESPONDER is 4.

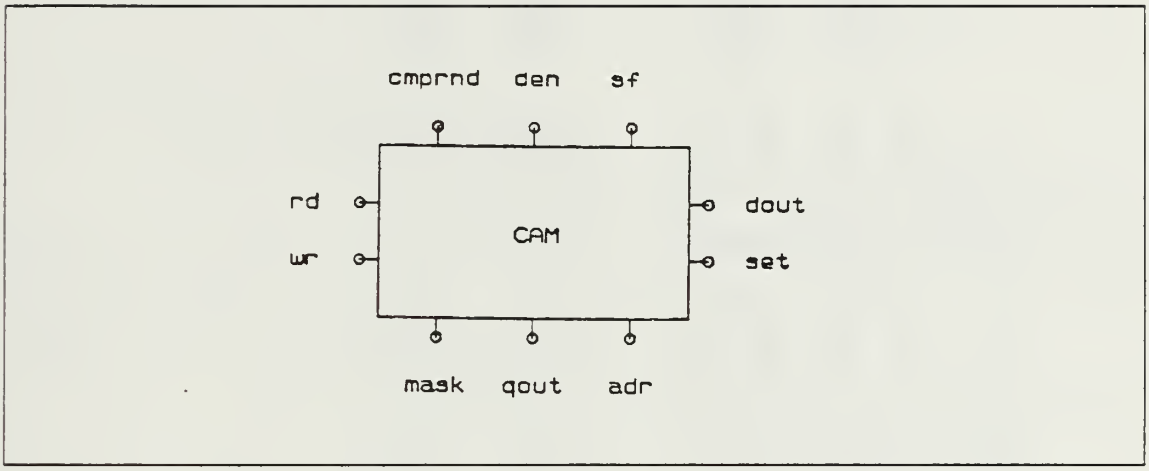


Figure 2.9 CAM System Body.

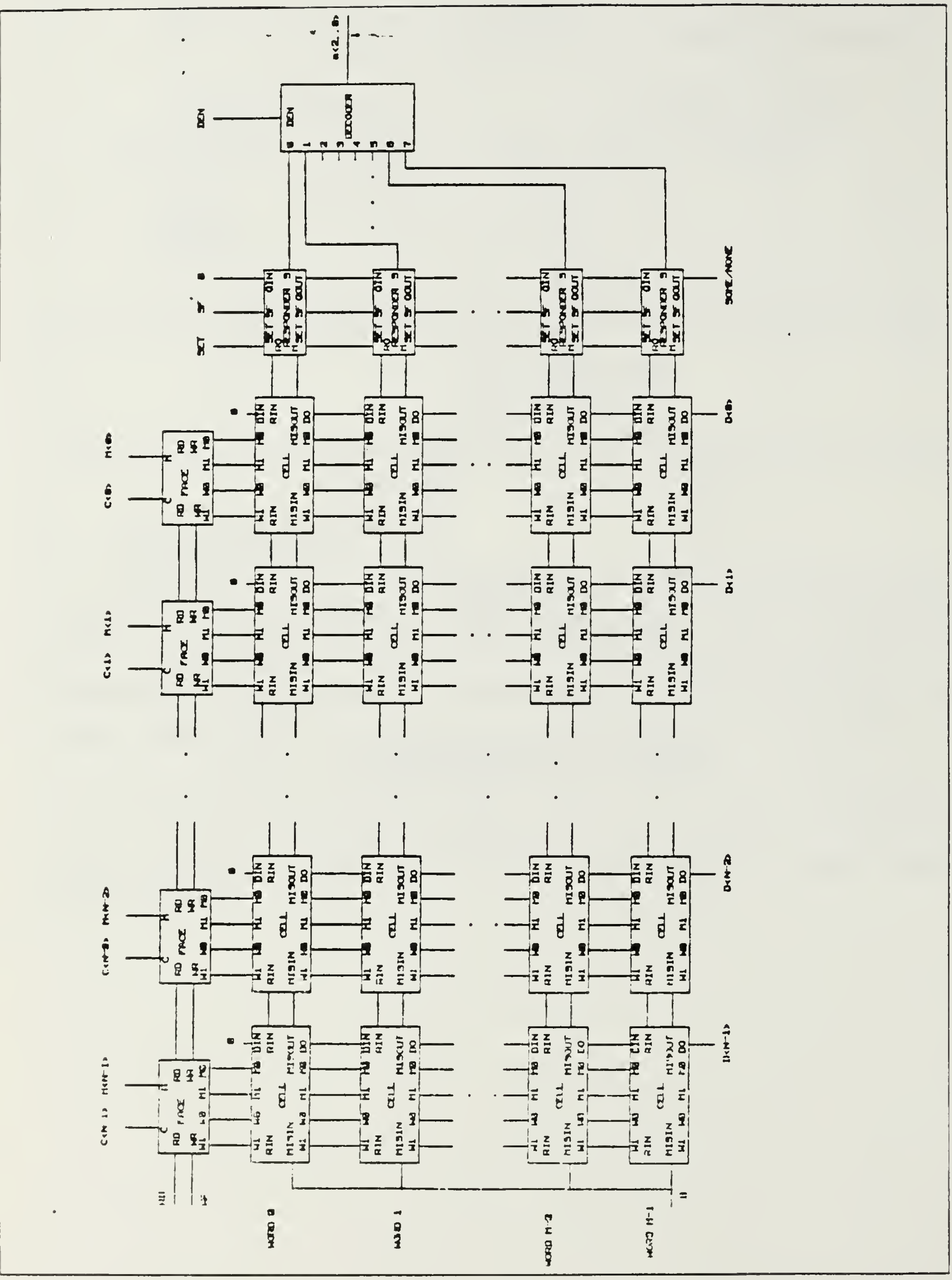


Figure 2.10 CAM System Diagram.

III. CAM PERFORMANCE ANALYSIS

In this Chapter, the CAM system timing is analyzed. This step is necessary for designing a CAM controller in the following Chapter. The timing values discussed in this Chapter are extracted from the SCALD system Verification step. The whole CAM system timing values are discussed using the studied results. Then the CAM simulation is presented. The restrictions of the CAM system are discussed from the results of the CAM simulation.

A. CAM TIMING ANALYSIS

1. Write Cycle Timing Analysis

a. FACE Write Cycle Timing Analysis

As shown in Figure 2.6, 2-Inverter and 1-AND gate delay is needed to generate the masked signals in the FACE circuit. Even though the comparand and the mask are available, the outputs of the FACE are inactive until low WR or low RD is applied. The total delay time is measured as 52 ns in Figure 3.1.

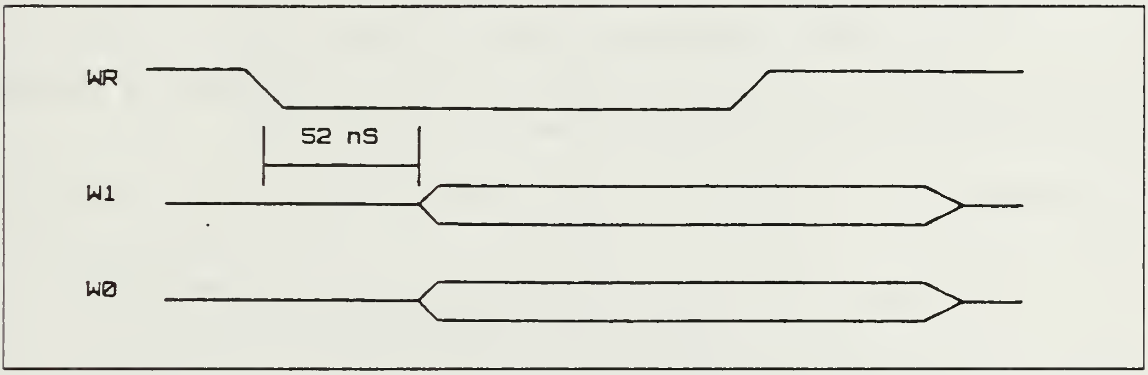


Figure 3.1 FACE Timing Diagram in the Write Cycle.

b. CELL Write Cycle Timing Analysis

To write information into the cell, the address has to be decoded in the DECODER, and R0 which is generated in the RESPONDER by the SELECT signal goes high. Information can not be written into the CELL until the RIN is high because the inputs of the RS Latch do not affect the contents of the CELL by keeping it low. The total delay time in the CELL after RIN is activated is measured as 47ns in Figure 3.2.

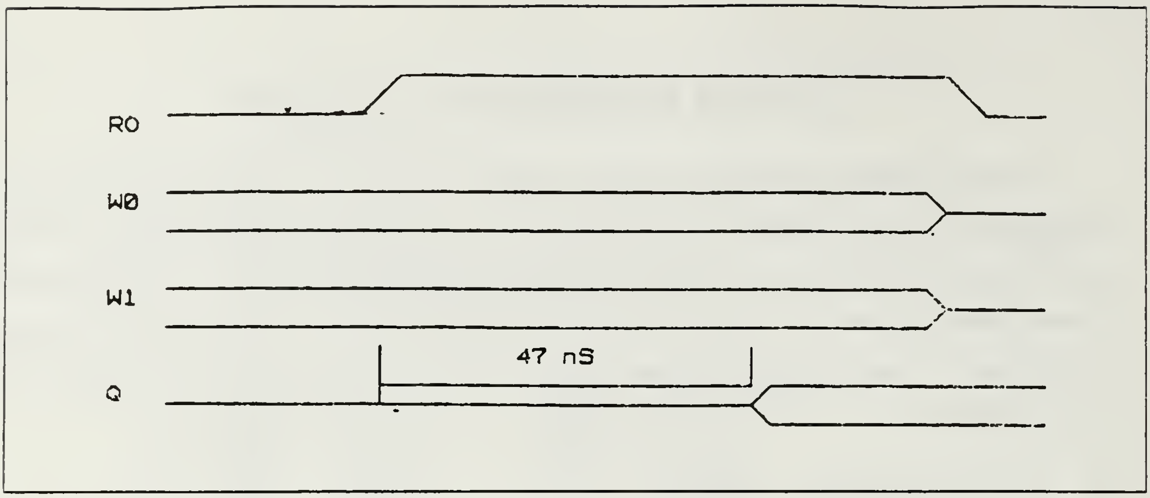


Figure 3.2 CELL Timing Diagram in the Write Cycle.

c. RESPONDER Timing Diagram Analysis in the Write Cycle

In the RESPONDER, the tag bit is turned on by receiving the high SELECT signal from the DECODER during the write cycle. The total delay time of the RESPONDER in the write cycle is measured as 58 ns. All other signals are inactive in this period and do not affect the tag bit.

d. DECODER Timing Analysis in the Write Cycle

Through LS138 and Inverter, the output of the DECODER is available at 52 ns after DEN goes low. The other inactive lines are zero.

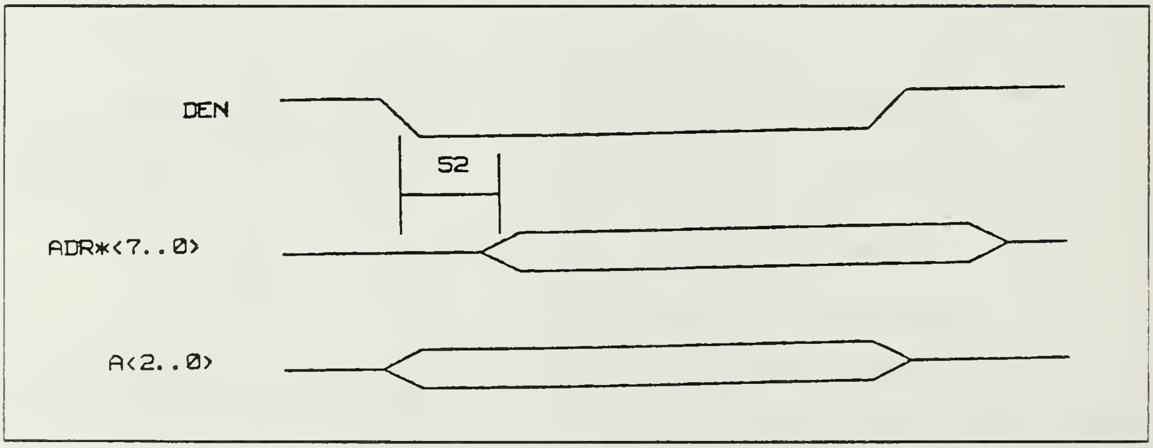


Figure 3.3 RESPONDER Timing Diagram in the Write Cycle.

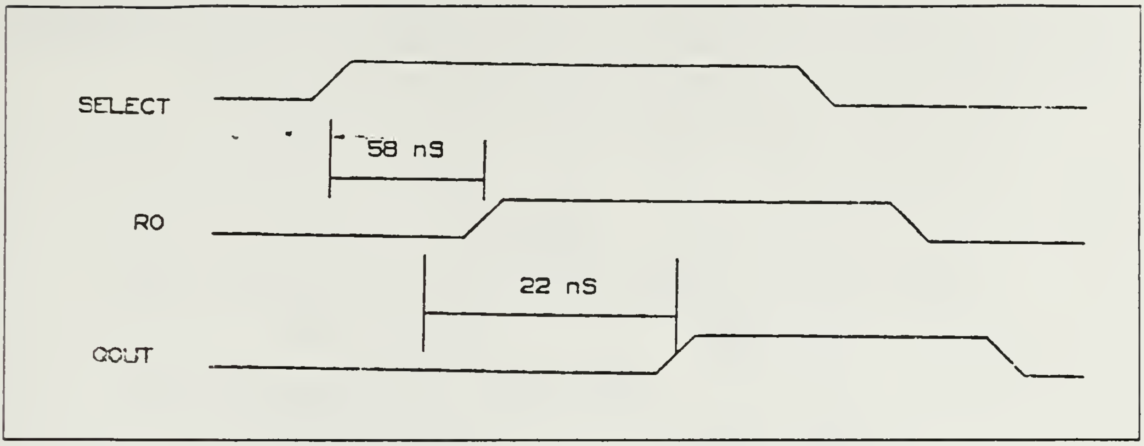


Figure 3.4 DECODER Timing Diagram in the Write Cycle.

TABLE 2

WRITE CYCLE TIMING DIAGRAM SYMBOLS AND TIME

SYMBOL	TIMING
TF	52
TS	52
TR	48
TC	47

e. CAM Timing Analysis in the Write Cycle

The CAM writing operation can be sequenced as follows:

a. First apply the address to the DECODER to produce the SELECT signal for the RESPONDERs.

b. Using the high SELECT signal, R0 becomes high in the RESPONDER and is passed to the row CELLs.

c. The FACE generates W0 and W1

d. Information is stored into the CELL using the above W1, W0 and RIN.

But as mentioned before, the masked W0 and W1 signal do not change the contents of the cell until RIN is active. So, step b) and c) or step a) and c) can be overlapped if the FACE enable signal (WR) is applied properly. As a result, the sequence of the CAM writing can be reduced to 3 steps. In this thesis, step b) and c) are overlapped and this overlapping relationship is explained in Table 3.

TABLE 3

WRITING SEQUENCE AND THE ACTIVE SIGNAL NAME

a	DECODER	ADDRESS, DEN
b	FACE	WR, MASK, COMPARAND, W1, W0
	RESPONDER	SELECT, R0
c	CELL	W1, W0, RIN

From Figure 3.5, the FACE circuit is activated with the RESPONDER circuit at the same time.

As shown in Figure 3.1 and Figure 3.3, the write cycle time of the FACE and the RESPONDER are 52 ns and 58 ns respectively. Even though the outputs of the FACE are available earlier than those of the RESPONDER, it does not cause any problems. This was explained in Chapter II. So, the CAM write cycle step can be reduced to three steps as shown in Table 3.

2. Read Cycle Timing Analysis

The reading operation can be sequenced as follows.

a. First, it turns on all tag bits by supplying the high SET.

b. By assigning RD, mask and comparand, the masked M0 and M1 signals are available and passed to the row CELLs.

c. Using M0 and M1 signals, the MISMATCH signal is activated and is passed to the RESPONDER through the row CELLs.

d. R0 in the RESPONDER is activated.

e. SELECT FIRST operation is operated if needed.

f. Information which is read from the CELLs are shown on the data bus.

In this read cycle, no steps are overlapped and all sequence described above are done sequentially.

a. FACE Read Cycle Timing Analysis

The outputs of the FACE, M0 and M1, are not available until the RD goes to low during the read cycle. The total read cycle delay time in the FACE is the same as the FACE write cycle time and was measured as 52 ns.

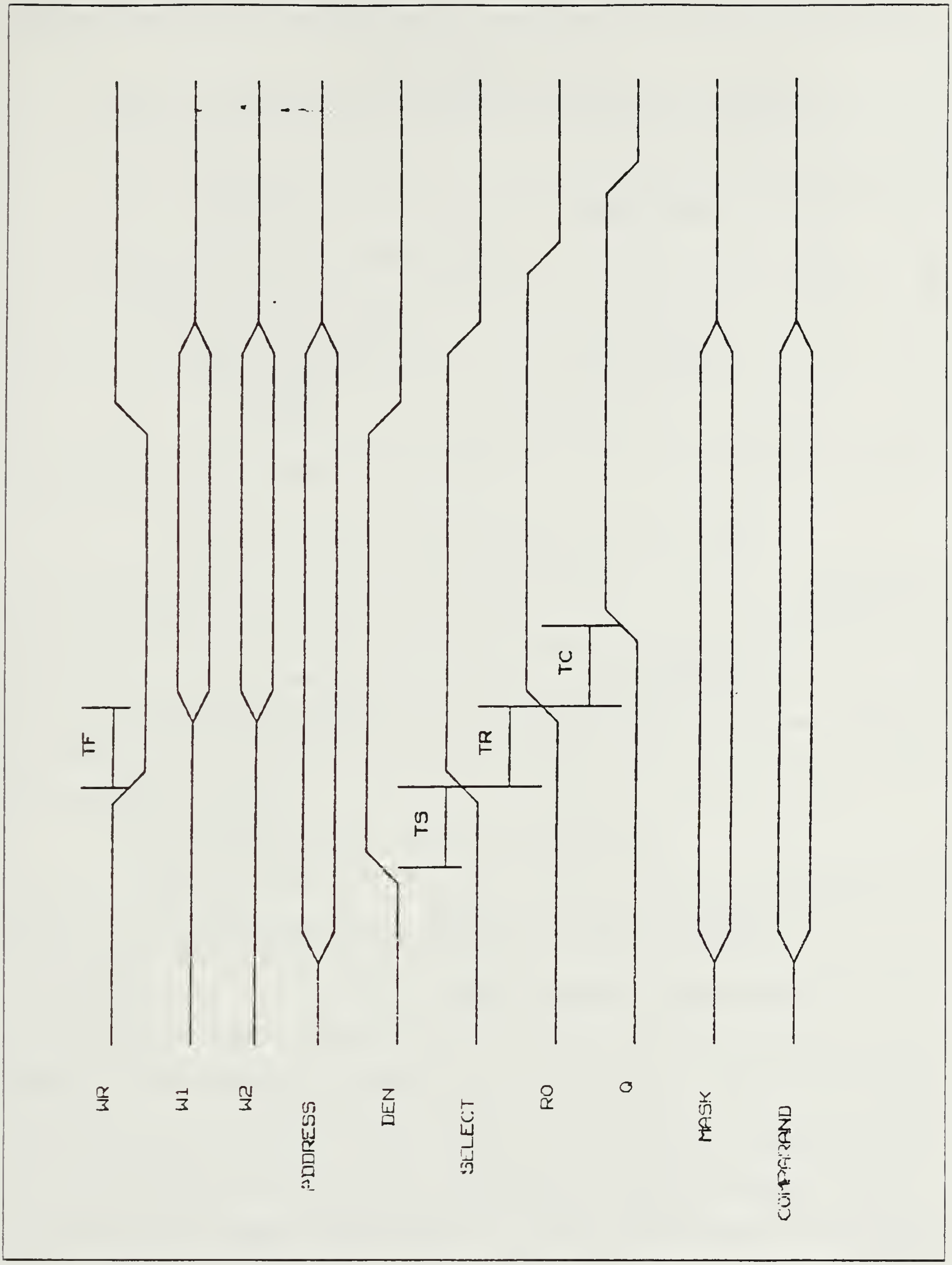


Figure 3.5 CAM Write Cycle Timing Diagram.

TABLE 4

READING SEQUENCE AND THE ACTIVE SIGNALS NAME

Step	Active Circuit	Active Signals
a	RESPONDER	SET, R0
b	FACE	RD, MASK, COMPARAND, M1, M0
c	CELL	M1, M0, MISMATCH
d	RESPONDER	MISMATCH, R0, QOUT
e	RESPONDER	SF, R0, QOUT
f	CELL	R0, D0

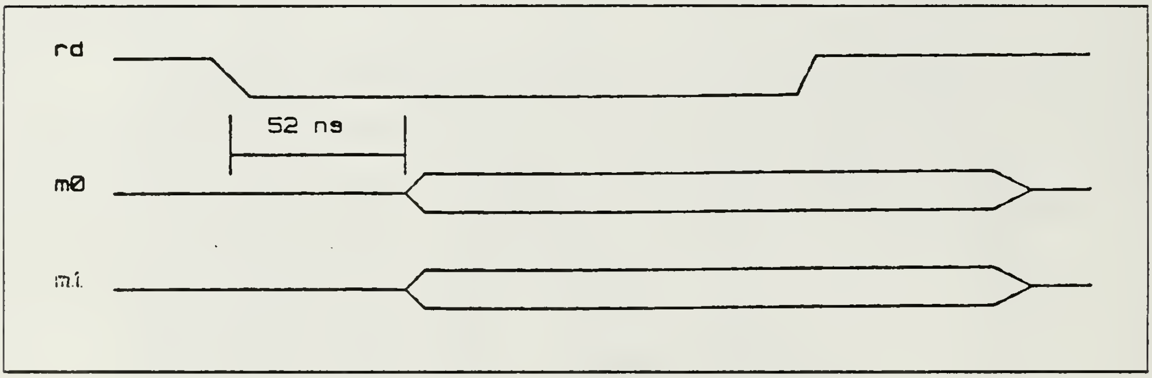


Figure 3.6 FACE Timing Diagram in the Read Cycle.

b. CELL Read Cycle Timing Analysis

In the CELL, if M0 and M1 are already masked in the face, the MISMATCH output is available after 52 ns through an AND, 3-Input NOR and an Inverter (15, 22 and 15 ns respectively) whether the RIN is available or not. The total MISMATCH delay time of the CAM system depends on the word length because the M0 and M1 signals are supplied simultaneously and the AND operation is done at the same time in the whole CELL circuits. As a result, the total delay time to produce the MISMATCH signal in the CELLs can be represented in the following equation.

Total MISMATCH delay time = $(15 + 37N)$ ns

where N represents the word length

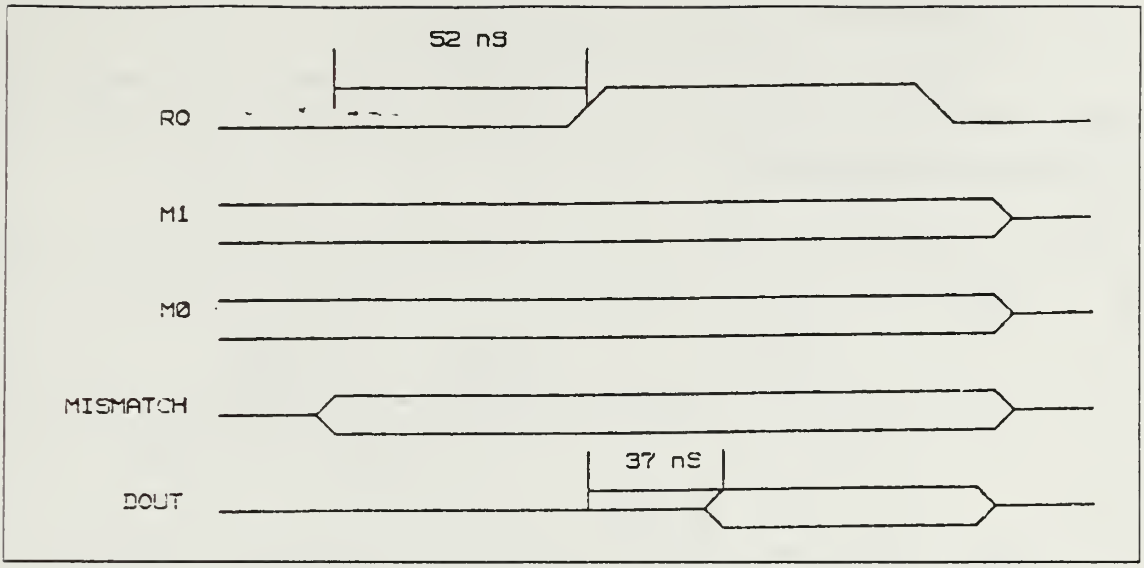


Figure 3.7 CELL Timing Diagram in the Read Cycle.

After RIN arrives from the RESPONDER, the DOUT is available on the data bus after 37 ns delay through AND gate 4P (15 ns) and OR gate 3P (22 ns). Also, the RIN is supplied simultaneously to the same row CELLs. The data output of the first word is not affected by the SELECT FIRST operation. So the total delay time required to show the data on the data bus is $15 + 22 \bullet (M - 1)$ ns, where M stands for the number of words.

c. *RESPONDER Timing Analysis in the Read Cycle*

Setting R0 to high takes at least 58 ns delay time by the SET signal. This operation is mandatory in the read cycle and is done simultaneously in the RESPONDER. During the read cycle, the MISMATCH is available in the CELL after the matching operation. It takes 64 ns delay time to clear the R0 if the MISMATCH is high, and takes 87 ns to clear the QOUT, the QOUT of the last stage is available through the OR gates and it takes $22 \bullet (M - 1)$ ns after the MISMATCH clears unmatched tag bits. At this time, the SELECT FIRST operation begins, and it takes about 90 ns delay to clear the tag bits. The QIN in the first stage RESPONDER has to be low because if QIN is high, it may be possible for the SELECT FIRST operation to clear all the R0 signals. And the first stage RESPONDER does not change its signal by performing the SELECT FIRST operation.

d. DECODER Timing Analysis in the Read Cycle

During the read cycle, the DECODER must keep in the inactive state because the SELECT signal is not required during the read cycle. This action is done by keeping the END signal low.

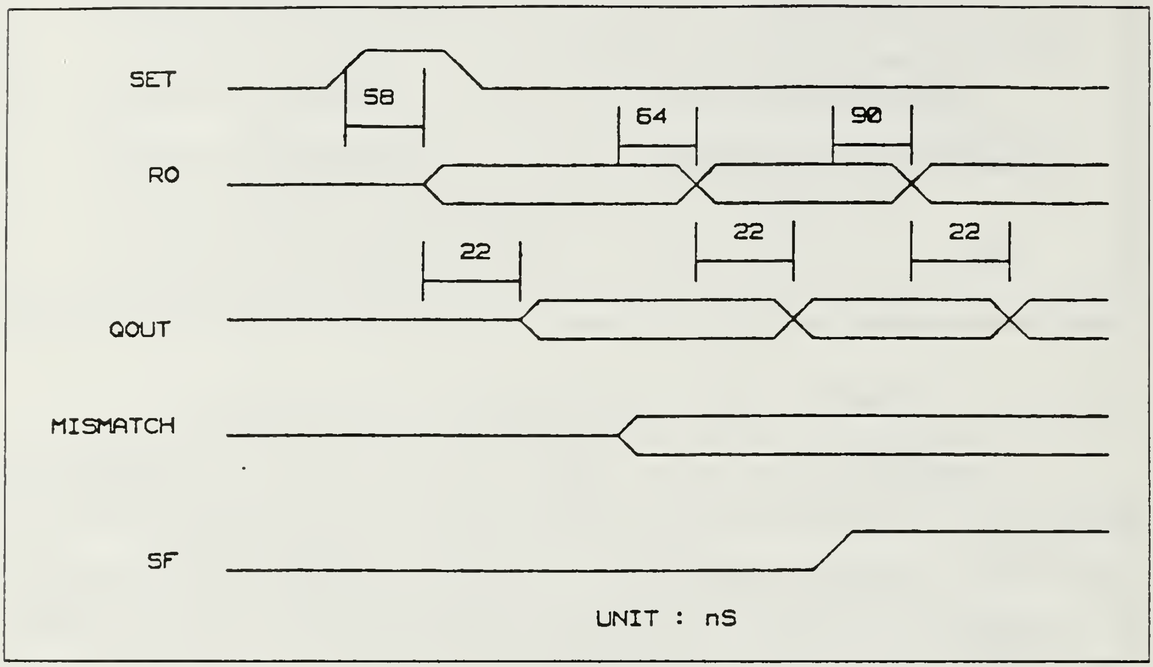


Figure 3.8 RESPONDER Timing Diagram in the Read Cycle.

e. CAM Timing Analysis in the Read Cycle

Figure 3.9 shows the CAM system read cycle timings. The timing values shown in Table 5 were extracted from the timing diagrams of all the blocks explained in the previous section.

All symbols are related to the read operation sequence except TQ which is also included in step d). The total time required to read the information from the CAM system is calculated by adding all timing values shown in Table 5.

Read Cycle Time = 272 + 44M + 37N

where M stands for the number of words and N stands for the word length.

B. CAM TIMING SIMULATION

One of the purposes for simulation is to check that the expected results are coming out of the circuit. For the same reasons, the CAM simulation is needed to

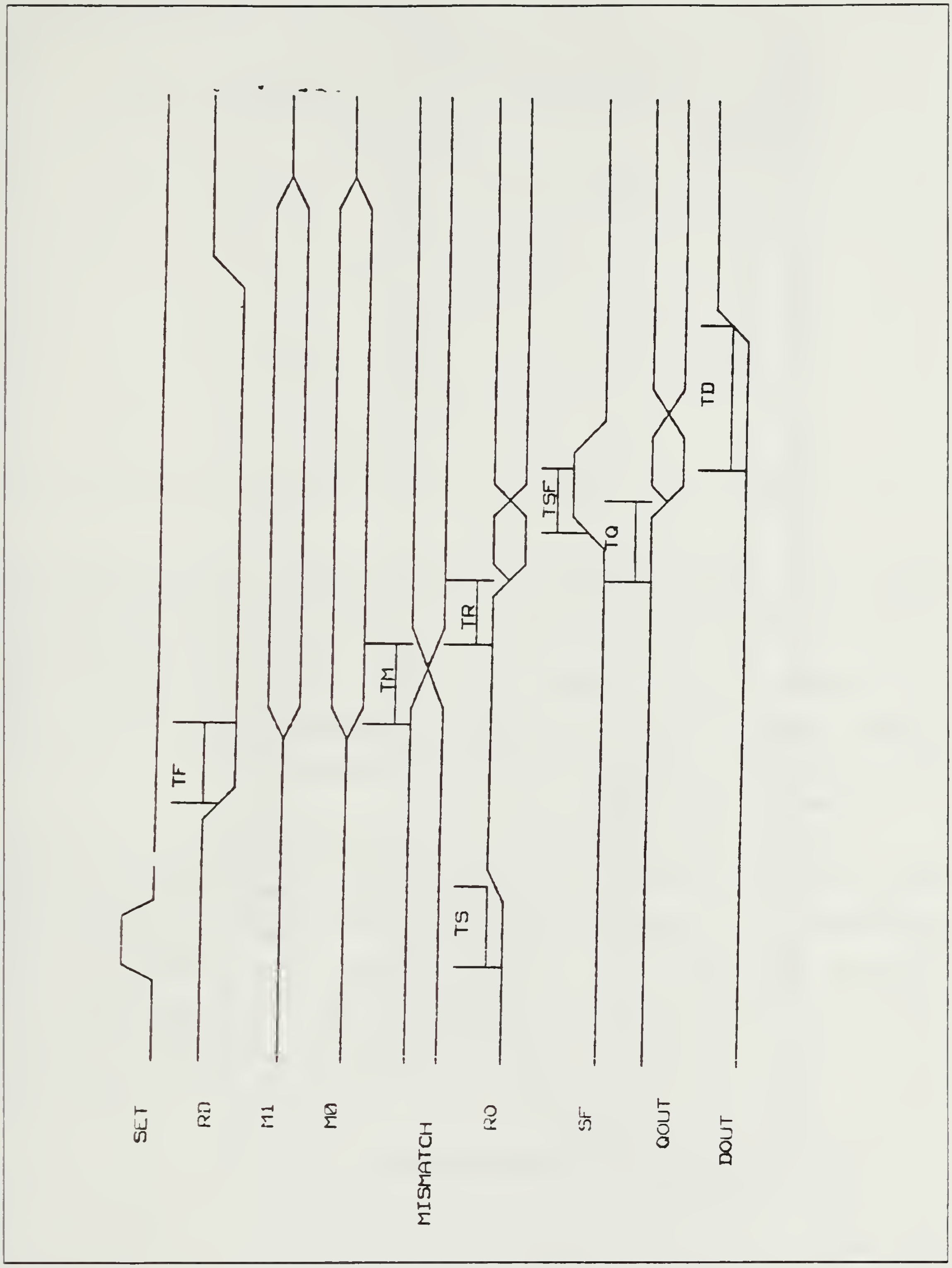


Figure 3.9 CAM Timing Diagram in the Read Cycle.

TABLE 5

READ-CYCLE TIMING DIAGRAM SYMBOLS AND TIME

SYMBOL	TIMING(ns)
TS	58
TF	52
TM	15+37N
TR	64
TQ	22M
TSF	90
TD	15+22(M-1)

ensure that the timing analysis done in the previous section is correct. Using those timing values which are shown in Table 3 and Table 5 by substituting M as 4 and N as 8, the four word CAM simulation is done to check that the timing analyses are correct. The simulation program for the SCALD system is listed in Appendix A. The waveform diagram of the SCALD system CAM simulation in both the write and the read cycles are shown in Figure 3.10 and Figure 3.11 respectively.

1. Four Word CAM Timing

All required timing values for the CAM writing and reading were described in Table 3 and Table 5. M becomes 4 and N equals 8 since a four word CAM is used in the simulation. The resulting timings for the CAM reading can be calculatedly substituting M as 4 and N as 8 in Table 5 and are shown in Table 6. The total required time for the CAM reading in calculated as 744 ns. The write cycle time is the same as that in Table 3 because the CAM writing time is independent on the size of the CAM. The following is the scenario of the CAM simulation. All numeric values are represented in hexadecimal digits.

a. Select the sample data as 11, 33, 77 and FF.
b. Write 11 to CAM 0.
c. Repeat step b) until FF is written into CAM word 3.
d. Read 11 from the CAM system.
e. Repeat step d) until FF is read from the CAM.

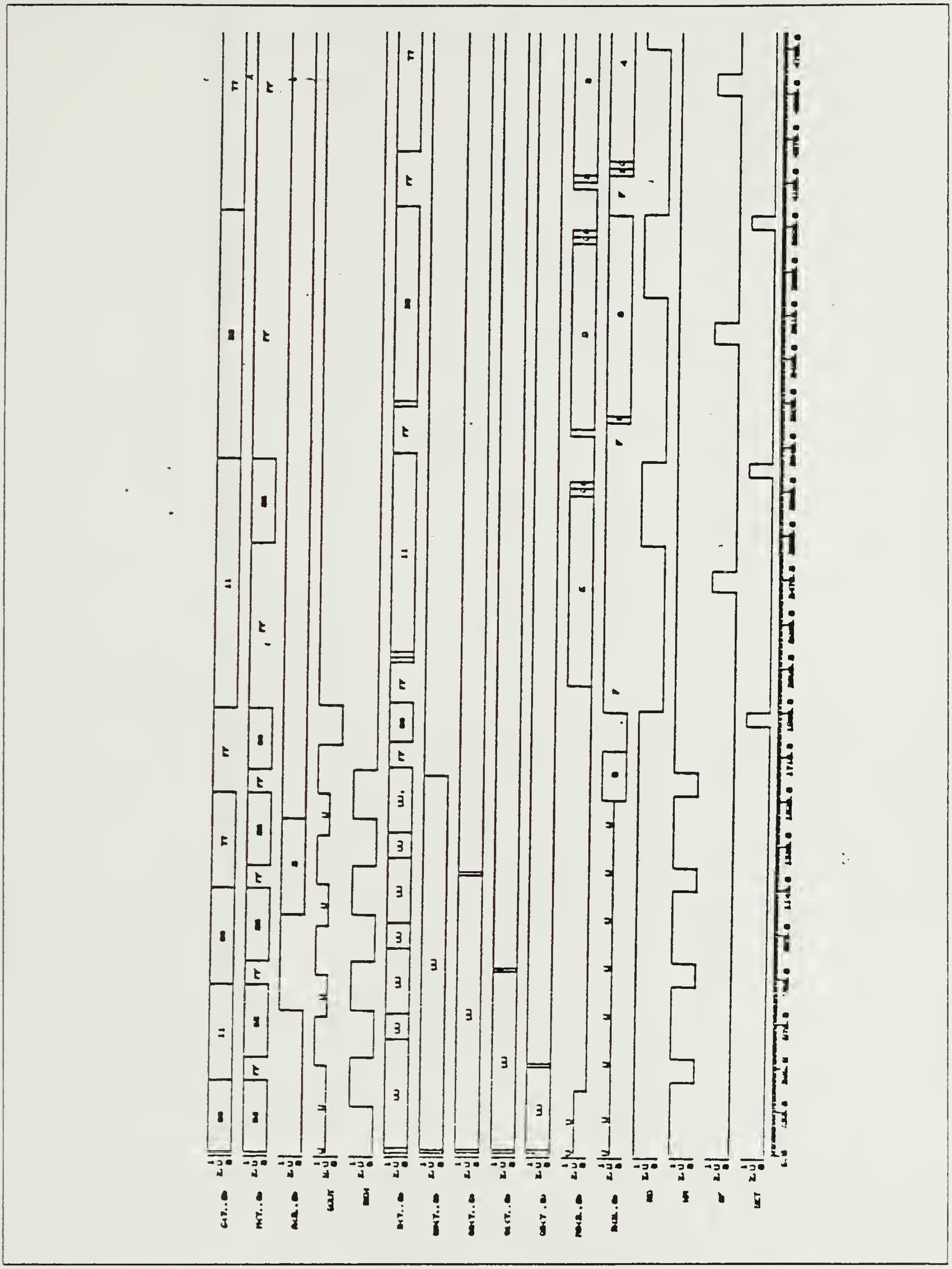


Figure 3.10 CAM Simulation Waveform Diagram.

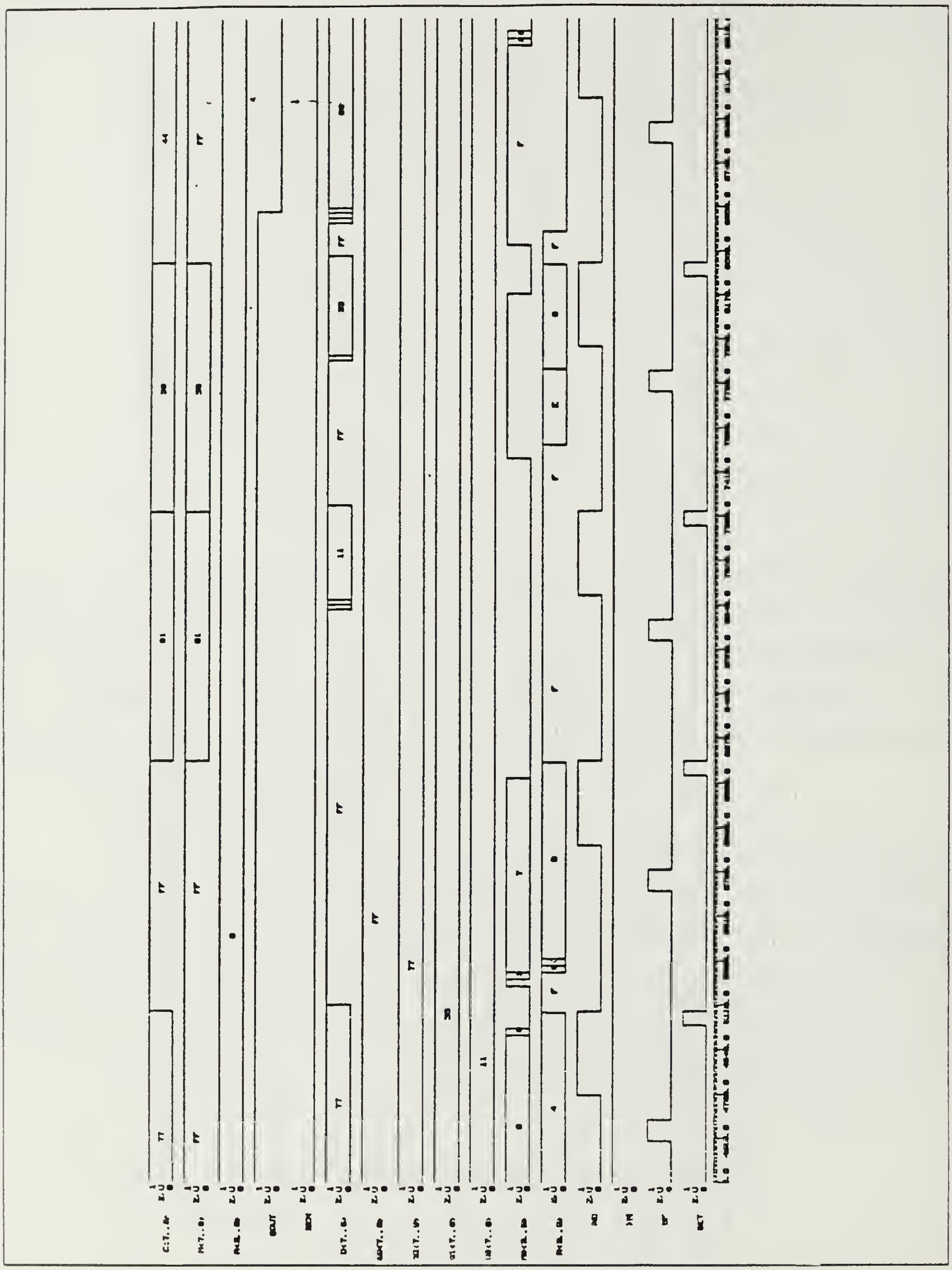


Figure 3.11 CAM Simulation Waveform Diagram (Continued).

f. Test partial matching with the sample data 01 and 30.

g. Do the unmatching test with the sample data 44.

TABLE 6
FOUR WORD READ CYCLE TIMING VALUES

SYMBOL	TIMING (ns)
TS	58
TF	52
TM	311
TR	64
TSF	90
TQ	88
TD	103

2. Throughput Discussion

The results of the CAM simulation with the timing values of Table 3 for writing, and Table 5 for reading did not show any problem and desired results appeared on the anticipated lines. In the CAM simulation program listed in Appendix A, 100 or 200 ns is added to the end of each step. This is necessary for the identification of each step. The CAM read cycle time is calculated based on the results of the simulation and those timing values appeared in Table 7. The more detailed throughput analysis will be discussed in Chapter V.

C. RESTRICTIONS OF THE CAM SYSTEM

As calculated before, the read cycle time and the write cycle time is 209 ns and $272+37N+44M$ ns respectively. Generally, there is no need to use special hardware, i.e., MMU, as is the case in the conventional memory system. The read operation of the CAM is the same as the searching operation of the conventional memory system. In the CAM system, it is necessary to consider the read and write cycle control methods differently because the read and the write cycle time are different and we can not use the CAM system in the usual way in the present commercial CPUs. Also the

TABLE 7

CAM READ CYCLE TIME BASED ON SIZE

SIZE(K)	TIME(ms)			
	N=8	N=16	N=32	N=64
0.25	11.8	12.1	12.7	13.9
0.50	23.1	23.4	24.0	25.2
1	45.6	45.9	46.5	47.7
2	90.7	91.0	91.6	92.8
4	180.8	181.1	181.7	182.9
8	361.0	361.3	361.9	363.1
16	1442.2	1442.7	1443.2	1444.4
64	2884.2	2884.5	2885.0	2886.2

SF signal generation is needed during the read cycle since the general purpose CPUs do not support any special control signals during the read cycle. The other problem is that the read cycle time of the CAM depends on the size of the CAM since the CAM read cycle time is a linear function of M and N. As M and N grow, all signal timings which were described before must be expanded for proper operation. Also, the basic structure of the CAM does not support sequential data reading. To use the CAM as a conventional memory which can do sequential read, CAM circuit modification is needed.

In a conventional memory system, multiple word searching is possible. This action may be difficult in the CAM system because the CAM system does not support the sequential read. So, only if the CAM is used for single word table lookup can it shows its full power.

All logic elements used in this thesis are TTL. So, if faster logic elements are used for the CAM realization, i.e., ECL, the total delay can be reduced significantly.

The total number of gates are significantly increased in the CAM system. Each CAM CELL needs two more gates than the conventional memory device [Ref. 1], and the extra FACE and RESPONDER circuits are needed. Even though hardware costs have decreased as VLSI techniques have developed, it is obvious that the CAM system costs much more than the conventional memory system because the CAM requires more gates.

There is no problem in the write cycle because the cycle time is independent of the memory size. In the conventional TTL memory system, the memory access time lies between 100 to 250 ns. If the same decoding circuitry used in the conventional memory system is used in the CAM system, the total memory access time is changed from 158 ns to 306 ns by adding 58 ns used in the RESPONDER to set the R0 signal. So, the writing cycle in the CAM can be adapted to any CPU without any problems. As a summary, the restrictions of the CAM designed here are expressed in the following statements.

- Special hardware is needed to control a CAM because the read and the write cycle time of the CAM are different if the CAM is used in a digital system whose read and write cycle times are the same.
- SF signal generation is required.
- The CAM system does not support the sequential memory read like instruction fetching in the conventional memory.
- Multiple word searching is very hard.
- CAM requires more gates than the conventional memory system and it makes the CAM system expensive.

IV. CAM CONTROLLER DESIGN

In Chapter III, the CAM timing analysis was done and we found that some special method is needed to control a CAM system due to the timing difference between the write and the read cycle and the size dependent read cycle time. In this Chapter, the CAM control methods and the design considerations are discussed. A CAM controller is designed to simulate the whole CAM system in the following Chapter. To use the CAM system correctly, the following methods may be considered.

- Expand the memory access time using NOP instruction in microprocessors. The SF signal generation is still required.
- Use special hardware, like a bit-slice microprocessor, as a CAM controller.
- Design special hardware which contains the SF generator and the WAIT state generator to expand the memory read cycle time with the CPU.

The first method is not attractive because using prolonged machine cycle instructions is wasteful. The second method may be a good one if the CAM size is large but it costs too much to develop a microprogram for a controller. If the CAM size is relatively small, the last method may be chosen because it can be designed in relatively short time and it costs little. A circuitry for communication may be required in the second and third methods to exchange signals between the processor and the CAM system.

In this thesis, the CAM controller methods is chosen among the several alternatives because the CAM size is small. Also, the standard bus in a PROLOG system is used in the comparison for timings. The PROLOG system has a 4 MHz Z80A microprocessor and the system clock operates at 4 MHz [Ref. 4]. So, one T state is 250 ns. The CAM controller is designed to control a four word CAM with a word length of eight bits. All necessary timings in the CAM are adapted to the standard bus timing. The PROLOG system timing diagram are posted in Appendix C.

A. DESIGN CONSIDERATIONS FOR CAM CONTROLLER

The followings are the considerations for the CAM system controller design in this thesis.

- Calculate the required timing values for the CAM read cycle.
- Determine the number of WAIT states and design the WAIT state requester.
- Design and generate the controller circuit including the SF generator and other control signals like output enable(DOEN), write enable, and DECODER enable(DEN).

B. CAM CONTROLLER DESIGN

From the PROLOG system memory read cycle timing diagram, RD* goes low at 35 ns after the MEMRQ* is active. We can not do any operation until RD* and MEMRQ* are available. RD* stays low during the next 405 ns [Ref. 4]. As a result, the CAM read operation can not be done in a standard memory read cycle because the CAM requires at least 744 ns delay. We need to generate WAIT back to the CPU to expand the memory read cycle time. Because the CPU operates at 4MHz, we need at least 2 WAIT states. RD* can stay low for 905 ns by adding two T states (500 ns). Also, the SF signal must be generated at least 418 ns after RD* is active and stay high during the next 156 ns. The TS used for setting the tag bit to high in the RESPONDER is needed in the beginning of the read operation. In this thesis, the SET operation is done separately. This means that an extra instruction, like OUT in Z80 microprocessor, is used for the SET function. So, the total CAM read cycle time is reduced to 686 ns by subtracting 58 ns from 744 ns. Here are the controller signal names and their functions:

- DEN -- Decoder Enable signal. This signal is activated in the write cycle only. During the read cycle, it is disabled to low. It is active high.
- MCEN -- Mask and Comparand register Enable signal. During the read or write cycle, the mask and the comparand register hold information and pass it to the CAM when enabled. It is active high.
- SF -- Select First signal.
- WAIT -- generate a WAIT signal to the microprocessor. It is active low.
- DOEN -- Output Enable signal. It becomes active in the read cycle only. It is active high.

In the standard bus, the WAIT signal hold time before T2 goes low is at least 90 ns [Ref. 4] and the WAIT hold time for the controller is 100 ns. For the second WAIT state generation, low WAIT stays until the falling edge of T3 state (actually Tw state). This time is measured as 415 ns and the second WAIT hold time is solved automatically. The SF signal is generated at 570 ns after RD* goes to low, and stays high during the next 125 ns while four word CAM requires 509 ns and 90 ns relatively. This slight over tolerance for the timing values may not cause any problems because all timing values described before are minimum time. If the higher frequency is used, we can get more precise controller timing even though the circuit of the controller become more complex. From Table 8, the WAIT signal is the same as the reverse of the Q4 state. To make MCEN, MEMRQ* is inverted and supplied to the mask and the comparand register during the memory read or write cycle. This is necessary because if it is activated during the write cycle, the input data may be mixed with the meaningless CAM output and becomes garbage. Also, the DOEN signal realization is achieved by combining RD* and MEMRQ* because the data output is available in the read cycle only. Table 9 shows the DEN signal truth table used in the memory write cycle.

The 8 Mhz clock is supplied to the D Flip Flop. In Figure 4.1, the fifth D Flip Flop is activated if the DOEN is high (memory read cycle).

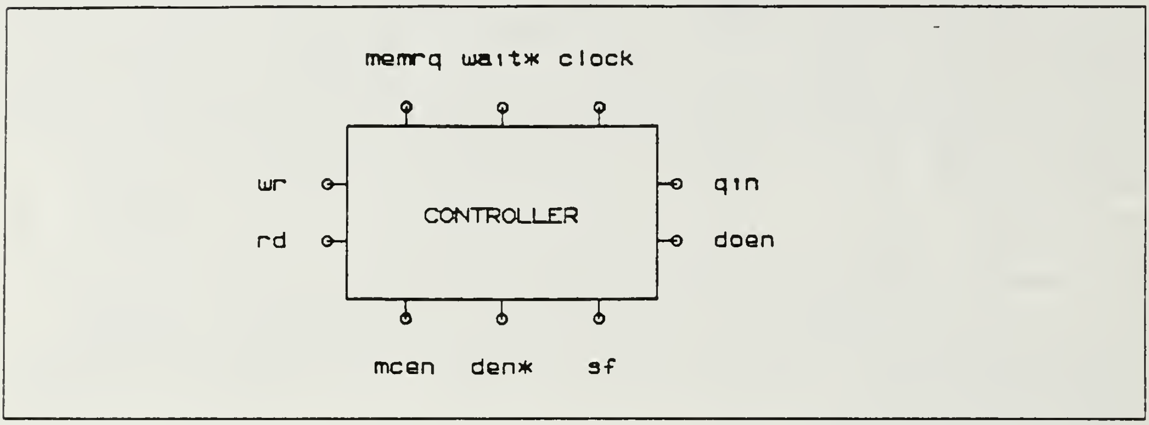


Figure 4.1 CAM Controller Body.

This is necessary for a generalized controller design because we can not guarantee that QOUT will be high or low at some specific time. So, the signal which is labelled as SF depends on the SOME/NONE signal which coming from the RESPONDER. When QOUT becomes the same as SOME/NONE activated, SF is high because DOEN is already turned on. If QOUT is low, SF is never turned on even though the fifth Flip Flop is high. From Table 8, Q4 state becomes high once again during the read cycle. But, this active WAIT signal does not affect the generation of the new WAIT states because the microprocessor detects the WAIT request at the falling edge of the T2 state. Also, this WAIT signal is transferred to the last D Flip Flop because the clock operates all the time. To prevent unwanted SF signals from appearing in any other machine cycle, the output of the fifth Flip Flop is ANDed with the DOEN. So, the delayed SF signal will never show up in the other machine cycle. It will be activated only in the read cycle.

TABLE 8

TRANSITION TABLE OF CONTROLLER

PRESENT STATE					NEXT STATE				
Q4	Q3	Q2	Q1	Q0	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	1	1	0	0	0
1	1	0	0	0	1	1	1	0	0
1	1	1	0	0	1	1	1	1	0
1	1	1	1	0	0	1	1	1	1
0	1	1	1	1	0	0	0	0	1
0	0	0	0	1	1	0	0	0	0
1	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0

C. THROUGHPUT DISCUSSION

This CAM controller was designed to control a four word CAM. If the word length and the number of words becomes large, almost the whole circuit should be modified for the correct timing except MCEN*, DEN* and DOEN generation circuitry. So, for a general system, i.e., in which the CAM size can be increased whenever the user wishes, a more generalized controller design procedure should be developed. As mentioned before, even though all timing values are satisfied to the desired specifications, the output timing value approximates of the controllers are still rough. If a faster clock is used as the Flip Flop driver clock, we can get more precise output timing values.

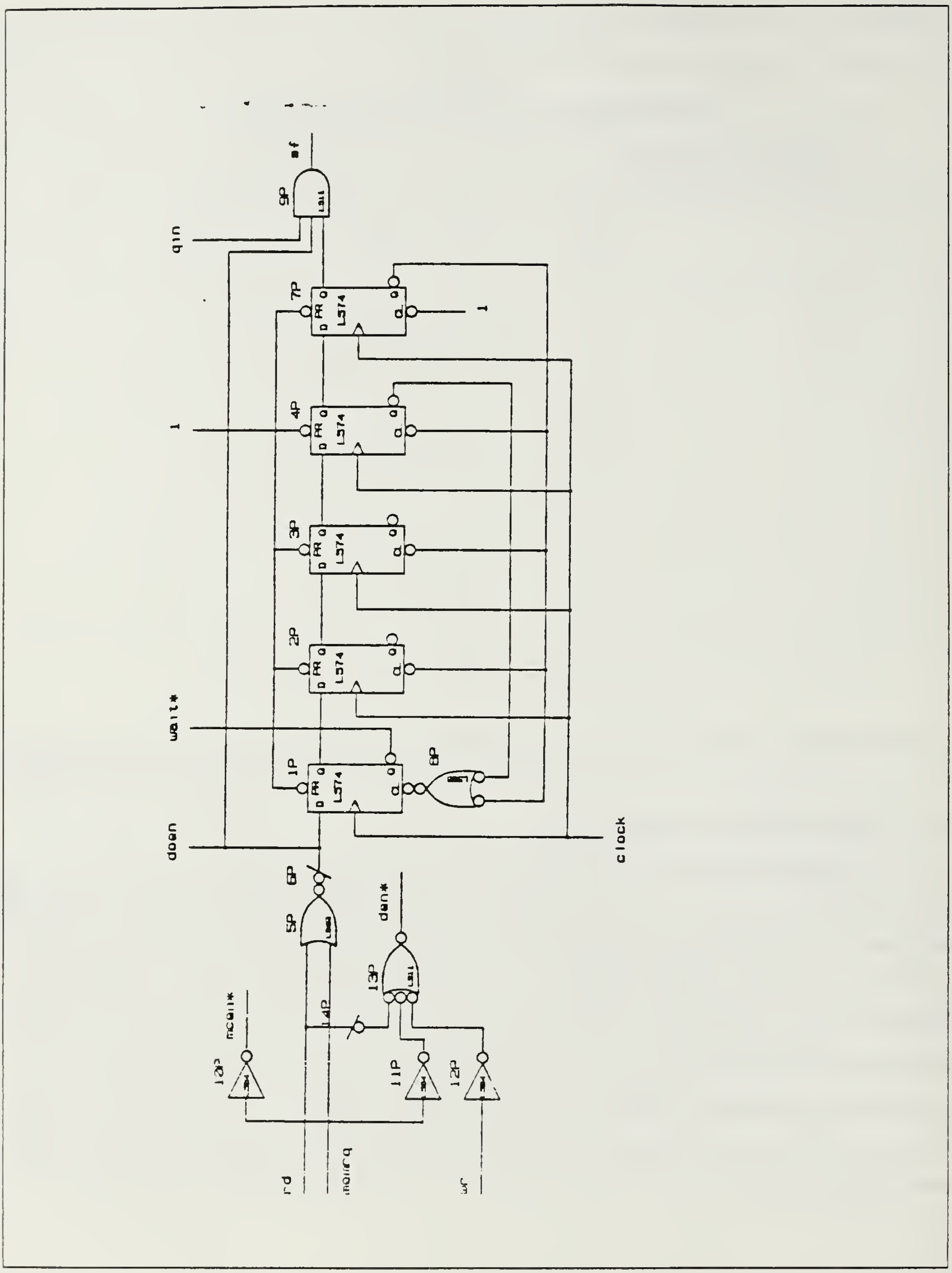


Figure 4.2 CAM Controller Circuit.

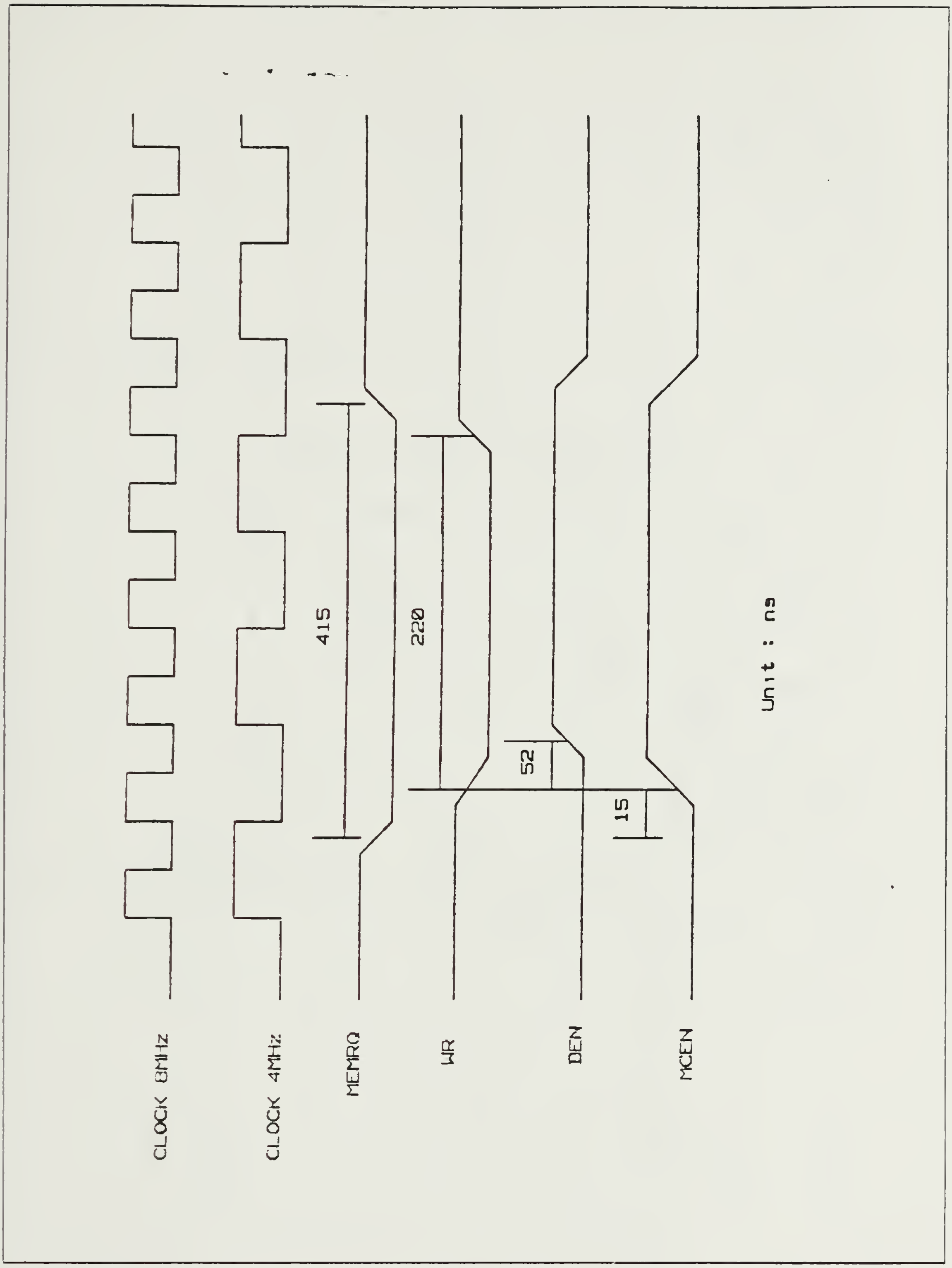


Figure 4.3 Controller Write Cycle Timing Diagram.

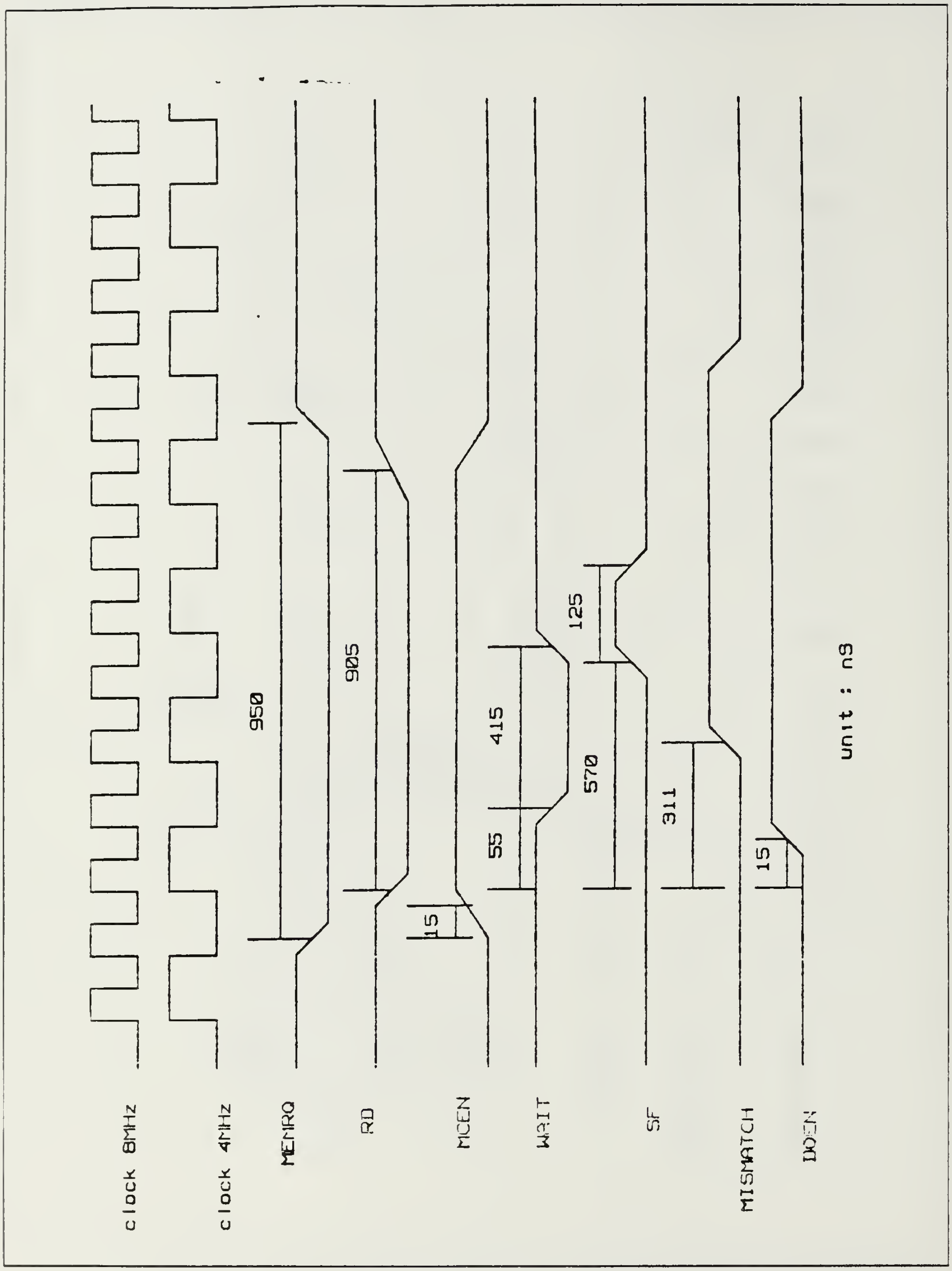


Figure 4.4 controller Read Cycle Timing Diagram.

TABLE 9

TRUTH TABLE FOR SIGNAL DEN

RD*	WR*	MEMRQ*	DEN	REMARK
0	0	0	0	Never happen
0	0	1	0	Never happen
0	1	0	1	Read cycle
0	1	1	0	Never happen
1	0	0	0	Write cycle
1	0	1	0	Never happen
1	1	0	0	Read or Write
1	1	1	0	Usual state

V. CAM TIMING SIMULATION WITH THE CONTROLLER

In this Chapter, the CAM simulation with the controller designed in the previous Chapter is done. The purpose of the simulation is to check the functionality of the CAM interfaced to the microprocessor. Using the results of the simulation, the detailed throughput analysis is discussed.

A. CAM SIMULATION WITH CONTROLLER

All required circuits including the CAM controller and necessary timing analyses were done in Chapter II through Chapter IV. Using those circuits, the final CAM simulation is done. All necessary timing values are extracted from the results of the CAM timing calculation done in Chapter III and the PROLOG system timing diagram [Ref. 4]. The mask and the comparand register are addressed by the output port 0 and 1 respectively. To set the RESPONDER at the beginning of the read cycle, an OUT instruction in the microprocessor is simulated, but it can be implemented by another port, such as port number 2. Here is the simulation scenario.

The sample data, 11, 33, 77 and FF in hexadecimal are selected and stored to CAM words 0, 1, 2 and 3. To write the data into the CAM, the mask register is filled with FF and the comparand register is set with the sample data 11. After setting up the mask and the comparand register, the write operation is done by assigning 0 to the address bus and simulating the microprocessor memory read cycle. These steps are repeated until the last data FF is written into CAM word 3. To read the data from the CAM system, first the SET operation is simulated. After assigning a sample data 11 to the comparand, and FF to the mask register, the microprocessor memory read cycle with 2 WAIT states is simulated. This read operation is repeated until FF is read from the CAM system. Finally, the partial matching is tested with the sample data 01 and 30.

Using the above sequence, the simulation was done and the SCALD system simulation program is listed in Appendix B. The waveform diagram of the SCALD system CAM simulation in both write and the read cycle are shown Figure 5.2 and 5.3 respectively.

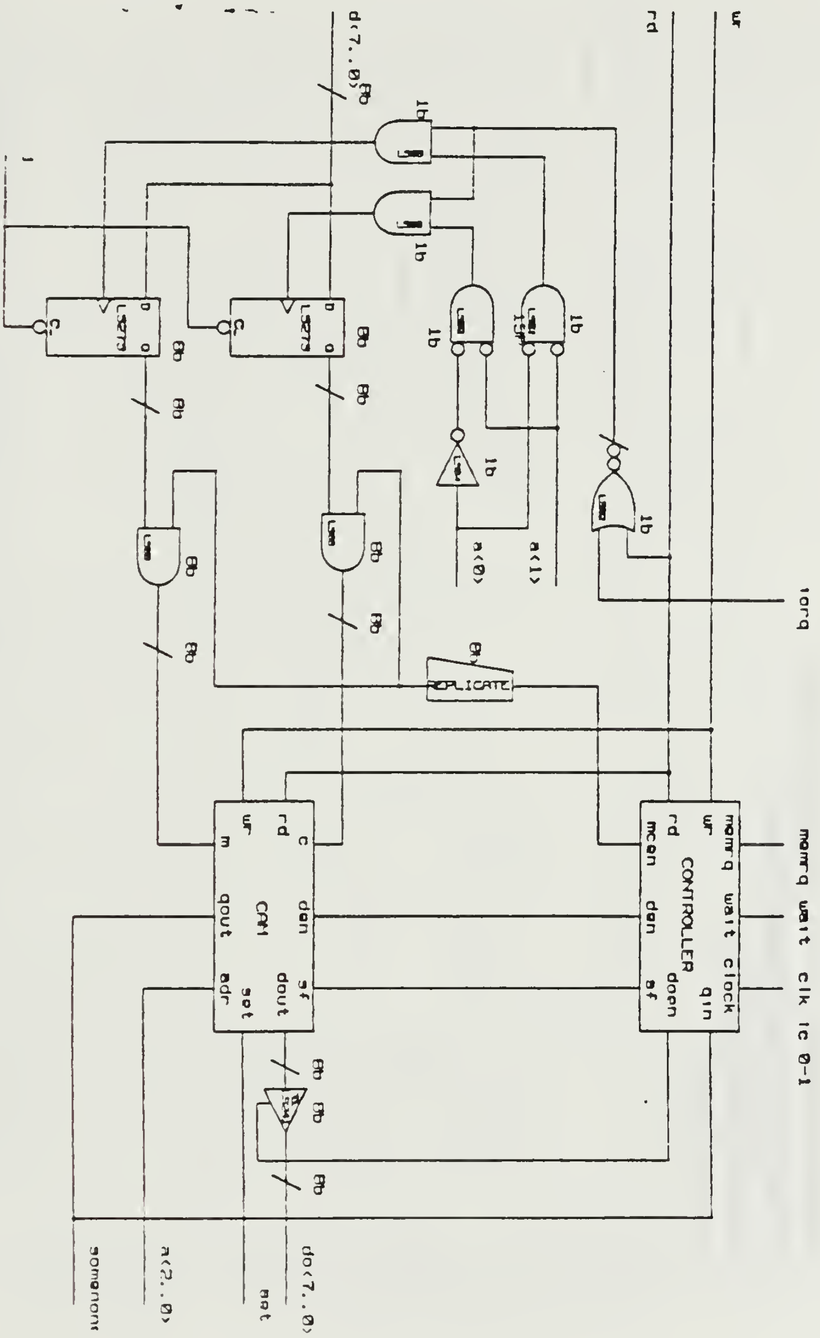


Figure 5.1 Simulation Circuit Diagram.

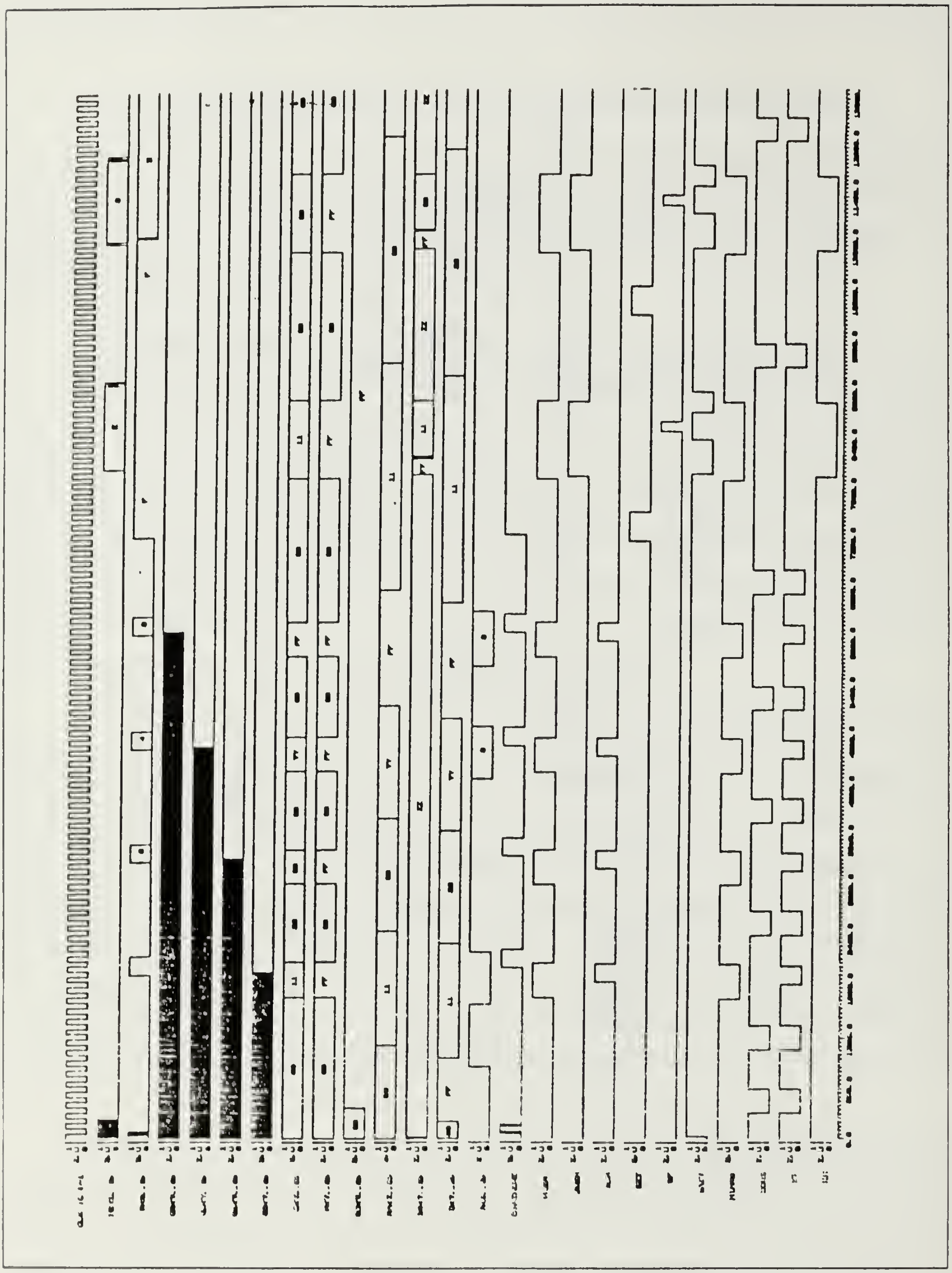


Figure 5.2 CAM with Controller Simulation Waveform Design.

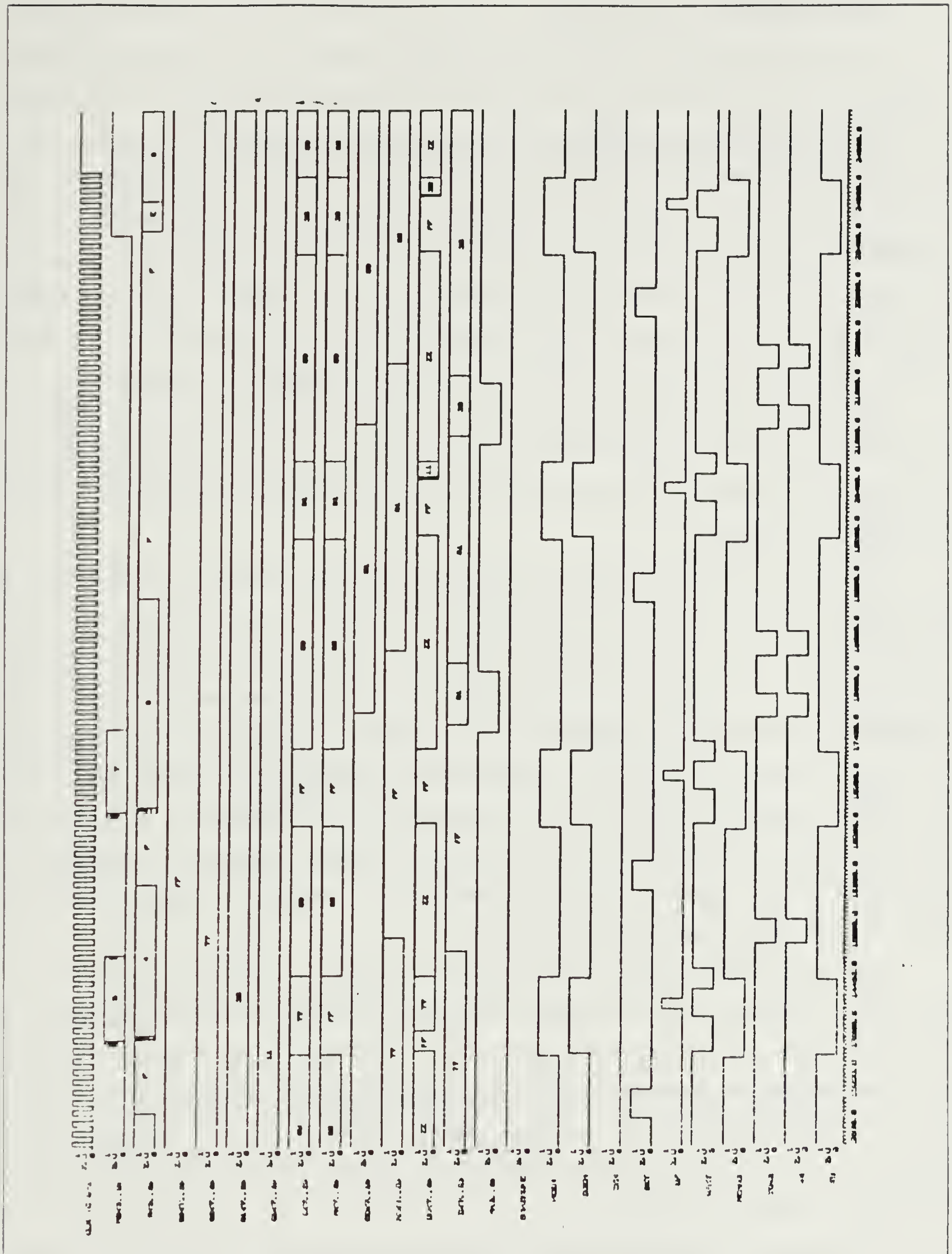


Figure 5.3 CAM with Controller Simulation Waveform Diagram (Continued).

B. THROUGHPUT ANALYSIS

To write information into the CAM system, we need to use at least one memory write operation. In addition, two more output instructions are required to fill the mask and the comparand register before the memory write operation. So, a total of three instructions are needed to write data into the CAM system. These additional two output instructions may become a significant problem in the CAM system. But, the additional two output instructions can be reduced to one output instruction by keeping the mask register as FF (HEX) during the write cycle. Table 10 shows the sample program for the conventional memory writing and Table 11 is provided for the CAM writing. The contents of the accumulator in the CAM program are meaningless.

The total required time to write data into the conventional memory system is calculated as $10+44M$ and $29+55M$ for the CAM system in T states [Ref. 4]. The write cycle time comparison between the CAM and the conventional memory is shown in Table 12 and Figure 5.4.

While the memory read cycle time of the conventional memory is independent of memory size, that of the CAM depends on CAM system size. However, the CAM system requires 2 output instructions to fill the mask and the comparand register before the read operation. These additional two output instructions can not be reduced because we can not guarantee that the contents of the mask or the comparand are always the same. Table 13 is a sample program for searching specific data in the conventional memory system. The total required time is calculated as $17+53M/2$ T states in average. Table 14 is a CAM system read operation simulation program. The contents of the HL register in the last instruction are meaningless because the CAM does not need address during the read cycle.

The total required time for the CAM reading is calculated as $101T + (568+44M)$ ns after the SET operation. Because the PROLOG system operates at 4 MHz, the total CAM reading time becomes $101 + (568+44M)/250$ T states. The read cycle time comparison between conventional memory (RAM) and the CAM is shown in Table 15 and Figure 5.5. All the above results, including the RAM reading and the writing, was extracted from the CAM simulation programs shown in Table 10 to Table 11 and from Table 13 to Table 14.

As a result, conventional memory writing requires less time than the CAM system since the RAM does not need output instructions while the CAM has significant time saving advantage over the RAM in the read cycle. This result is

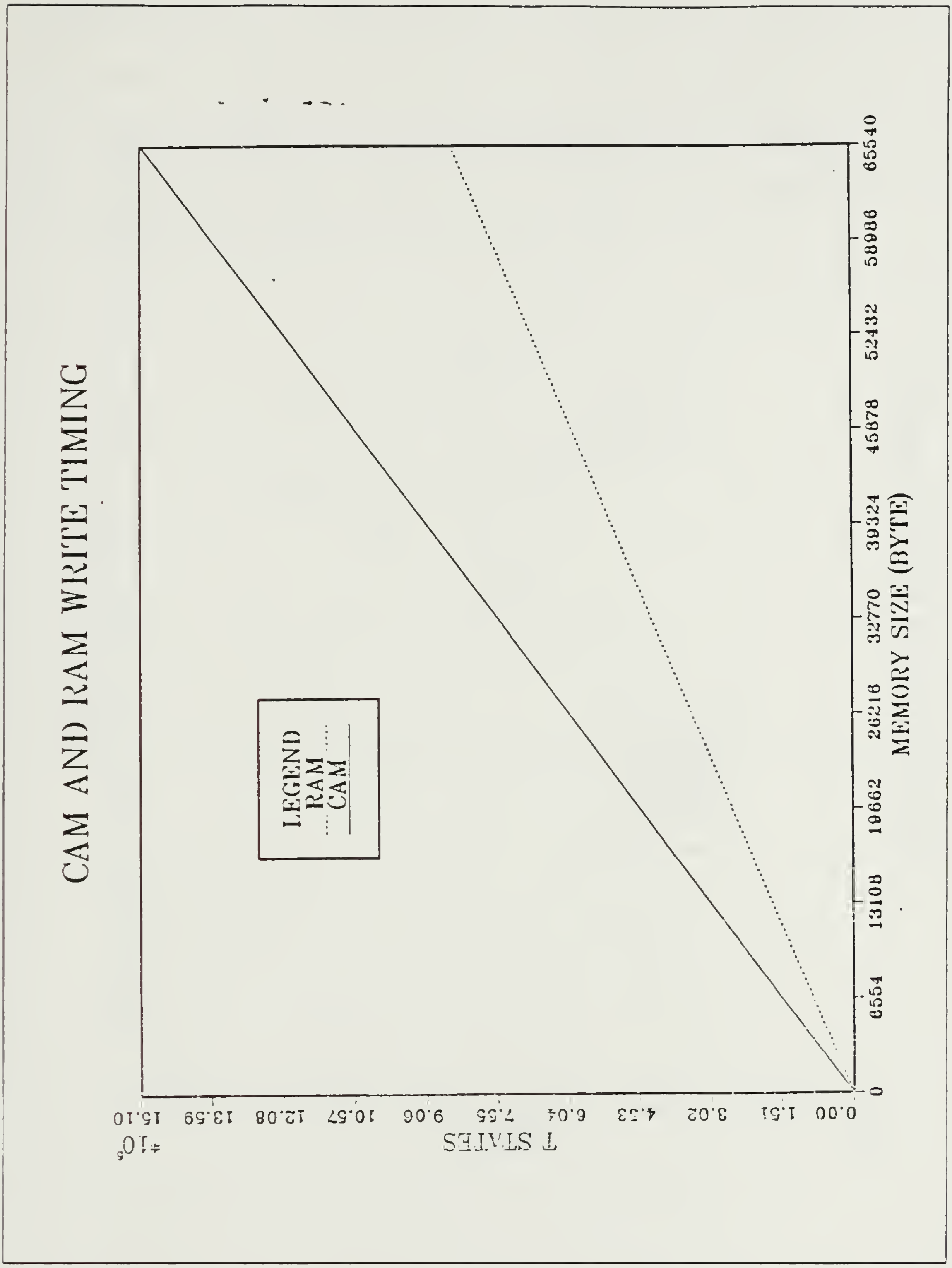


Figure 5.4 Comparison of CAM and Conventional Memory Write Timing.

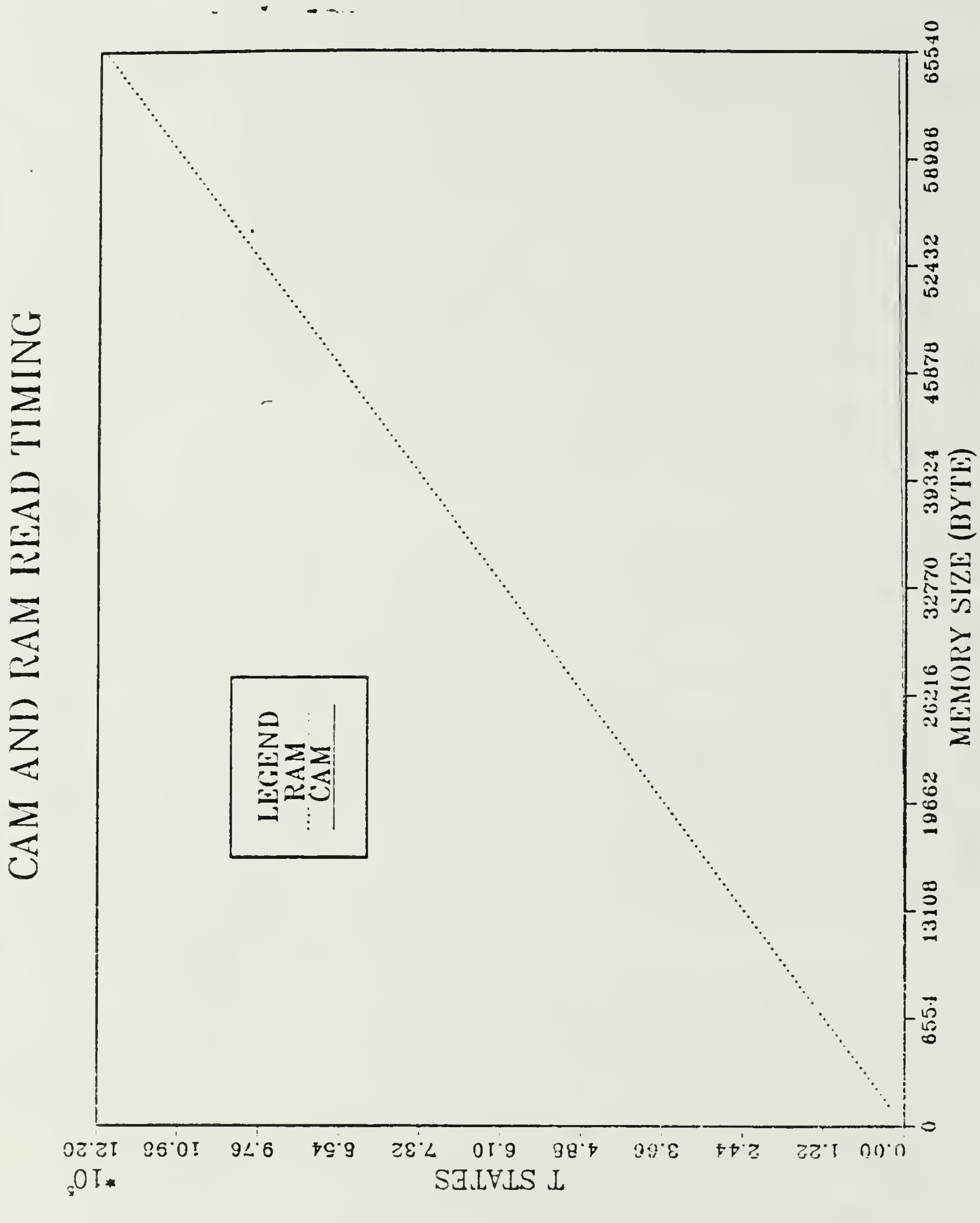


Figure 5.5 Comparison of CAM and Conventional Memory Read Timing.

natural because the data reading of the CAM is the same as the searching operation of the conventional memory. If more powerful controller is designed, the CAM read cycle time can be reduced more. As seen in Table 12, the CAM write operation takes almost one third more time than the conventional memory while the CAM read operation takes only 1/45 to 1/150 of the conventional memory reading time. Figure 5.4 and 5.5 were drawn using the results of Table 12 and Table 15.

TABLE 10

RAM WRITE PROGRAM FOR Z80

```
        LD      HL, MEMORYSIZE
LOOP :  LD      (HL), DATA
        DEC     HL
        JP      Z, DONE
        JP      LOOP
DONE:   . . . .
        . . . .
```

TABLE 11

CAM WRITE PROGRAM FOR Z80

```
        LD      HL, MEMORYSIZE
        LD       B, MASKDATA
        OUT   (MASK), B
LOOP:   LD       C, CMPRNDDATA
        OUT (CMPRND), C
        LD     (HL), A
        DEC        HL
        JP       Z, DONE
        JP         LOOP
DONE:   . . . .
        . . . .
```

TABLE 12

CAM AND CONVENTIONAL MEMORY WRITE TIMING

Memory Size(K)	Conventional Memory	CAM	Rate
0.25	10762	14109	1.317
0.5	21514	28189	1.310
1	43018	56349	1.3098
4	177042	225399	1.3096
16	688138	901149	1.3095
64	2752522	3604509	1.3095

TABLE 13

RAM READING PROGRAM Z80

```
          LD    B, DATA
          LD    HL, MEMORYSIZE
LOOP:     LD    A, (HL)
          CP    B
          JP    Z, DONE
          DEC   HL
          JP    Z, NOTFOUND
          JP    LOOP
DONE:     . . . .
          . . . .
NOTFOUND: . . . .
          . . . .
```

TABLE 14

CAM READING PROGRAM FOR Z80

```
            LD       A, 01
            OUT     (SET), A
            LD       BC, CMP_MASK_DATA
            OUT     (MASK), C
            OUT    (CMPRND), B
            LD       A, (HL)
            LD       D, 00
            SUB        D
            JP      NZ, FOUND
            XOR        C
            JP      NZ, NOTFOUND
    FOUND:  . . . .
            . . . .
 NOTFOUND:  . . . .
            . . . .
```

TABLE 15

CAM AND CONVENTIONAL MEMORY READ TIMING

Memory Size(K)	Conventional Memory	CAM	Rate
0.25	6806	149	45.64
0.5	13585	194	70.03
1	27153	284	95.61
4	108561	825	131.59
16	434193	2987	145.36
64	1736721	11638	149.23

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The objective of this thesis was to design a simple CAM system and a CAM controller. The following is the conclusions of this thesis.

The total number of gates required in the CAM system realization are much more than the conventional memory system. So the hardware cost for design a CAM system is more expensive than the conventional memory system. The read operation of the CAM system is the same as the searching operation of the conventional memory system. This characteristic of the system has a great advantage when a particular item is needed from a big data array. The CAM reading operation takes from 1/50 to 1/150 of the conventional memory read cycle time. The CAM, however, does not support the sequential read like the regular instruction fetch. This disadvantage makes multiple word searching difficult.

And the read cycle time of the CAM system is a linear function of the CAM size while that of the conventional memory is independent of memory size. The CAM write cycle operation does not have any problems when the CAM is used with the microprocessor if the comparand and the mask registers are provided. Because of this read and the write cycle time difference, some special methods to control a CAM system is required. The CAM controller selected among the alternatives and designed in this thesis worked without any problems. The faster clock will make it possible for the controller to generate more precise timing values. For a general system, a more generalized controller design procedure should be developed because of the read and the write cycle time difference in the CAM system.

B. RECOMMENDATIONS

The following recommendations will increase the performances of the CAM system:

- Using multi-input OR gates rather than 2-Input OR gates for transferring MISMATCH signal to the RESPONDER. It will decreased the CAM read cycle time since 8 OR-gates delay time will be reduced to one or two gates delay.
- Include the SET generator in the controller. It will decrease the SET operation from one instruction execution time to one quarter of a T state.
- Modify the CAM circuits or the controller circuit to read the information sequentially by assigning addresses.

- Modify the controller to use the input instruction rather than the memory read instruction for a CAM read if it is used in a microprocessor system. It will decrease the required CAM read cycle time.
- Realize the CAM circuit with the faster logic elements like ECL.

APPENDIX A
CAM SIMULATION PROGRAM FOR SCALD SYSTEM

```
    /* INITIALIZE */
WAVE      0   10000
HISTORY   20000
OPEN      WR
DEPOSIT   0
OPEN      RD
DEPOSIT   1
OPEN      SET
DEPOSIT   0
OPEN      SF
DEPOSIT   0
OPEN      M<7..0>;
DEPOSIT   00
OPEN      C<7..0>;
DEPOSIT   00
OPEN      QOUT
OPEN      DEN
DEPOSIT   0
OPEN      A<2..0>;
DEPOSIT   0
SIMULATE  100
    /* WRITE 11 INTO CAM WORD 0 */
OPEN      A<2..0>;
DEPOSIT   0
OPEN      DEN
DEPOSIT   1
SIMULATE  99
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   11
OPEN      WR
DEPOSIT   0
SIMULATE  105
OPEN      WR
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  200
    /* WRITE 33 INTO CAM WORD 1 */
OPEN      A<2..0>;
DEPOSIT   1
OPEN      DEN
DEPOSIT   1
SIMULATE  99
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   33
OPEN      WR
DEPOSIT   0
SIMULATE  105
OPEN      WR
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  200
    /* WRITE 77 INTO CAM WORD 2 */
OPEN      A<2..0>;
DEPOSIT   2
OPEN      DEN
```

```
DEPOSIT   1
SIMULATE  99
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   77
OPEN      WR
DEPOSIT   0
SIMULATE  105
OPEN      WR
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  200
     /* WRITE FF INTO CAM WORD 3 */
OPEN      A<2..0>;
DEPOSIT   3
OPEN      DEN
DEPOSIT   1
SIMULATE  99
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   FF
OPEN      WR
DEPOSIT   0
SIMULATE  105
OPEN      WR
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  200
     /* READ 11 FROM CAM SYSTEM */
OP        SET
DEPOSIT   1
SIMULATE  60
OPEN      SET
DEPOSIT   0
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   11
OPEN      RD
DEPOSIT   0
SIMULATE  515
OPEN      SF
DEPOSIT   1
SIMULATE  90
OPEN      SF
DEPOSIT   0
SIMULATE  103
OPEN      RD
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  300
     /* READ 33 FROM CAM SYSTEM */
OP        SET
DEPOSIT   1
SIMULATE  60
OPEN      SET
DEPOSIT   0
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   33
OPEN      RD
DEPOSIT   0
SIMULATE  515
```

```
OPEN      SF
DEPOSIT   1
SIMULATE  90
OPEN      SF
DEPOSIT   0
SIMULATE  103
OPEN      RD
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  300
     /* READ 77 FROM CAM SYSTEM */
OP        SET
DEPOSIT   1
SIMULATE  60
OPEN      SET
DEPOSIT   0
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   77
OPEN      RD
DEPOSIT   0
SIMULATE  515
OPEN      SF
DEPOSIT   1
SIMULATE  90
OPEN      SF
DEPOSIT   0
SIMULATE  103
OPEN      RD
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  300
     /* READ FF FROM CAM SYSTEM */
OP        SET
DEPOSIT   1
SIMULATE  60
OPEN      SET
DEPOSIT   0
OPEN      M<7..0>;
DEPOSIT   FF
OPEN      C<7..0>;
DEPOSIT   FF
OPEN      RD
DEPOSIT   0
SIMULATE  515
OPEN      SF
DEPOSIT   1
SIMULATE  90
OPEN      SF
DEPOSIT   0
SIMULATE  103
OPEN      RD
DEPOSIT   1
OPEN      M<7..0>;
DEPOSIT   00
SIMULATE  300
     /* READ 01 FROM CAM SYSTEM */
OP        SET
DEPOSIT   1
SIMULATE  60
OPEN      SET
DEPOSIT   0
OPEN      M<7..0>;
DEPOSIT   01
OPEN      C<7..0>;
DEPOSIT   01
```

```
OPEN       RD
DEPOSIT    0
SIMULATE   515
OPEN       SF
DEPOSIT    1
SIMULATE   90
OPEN       SF
DEPOSIT    0
SIMULATE   103
OPEN       RD
DEPOSIT    1
OPEN       M<7..0>;
DEPOSIT    00
SIMULATE   300
     /* READ 30 FROM CAM SYSTEM */
OP         SET
DEPOSIT    1
SIMULATE   60
OPEN       SET
DEPOSIT    0
OPEN       M<7..0>;
DEPOSIT    30
OPEN       C<7..0>;
DEPOSIT    33
OPEN       RD
DEPOSIT    0
SIMULATE   515
OPEN       SF
DEPOSIT    1
SIMULATE   90
OPEN       SF
DEPOSIT    0
SIMULATE   103
OPEN       RD
DEPOSIT    1
OPEN       M<7..0>;
DEPOSIT    00
SIMULATE   300
     /* READ 44 FROM CAM SYSTEM */
OP         SET
DEPOSIT    1
SIMULATE   60
OPEN       SET
DEPOSIT    0
OPEN       M<7..0>;
DEPOSIT    FF
OPEN       C<7..0>;
DEPOSIT    44
OPEN       RD
DEPOSIT    0
SIMULATE   515
OPEN       SF
DEPOSIT    1
SIMULATE   90
OPEN       SF
DEPOSIT    0
SIMULATE   103
OPEN       RD
DEPOSIT    1
OPEN       M<7..0>;
DEPOSIT    00
SIMULATE   300
```

APPENDIX B

CAM WITH CONTROLLER SIMULATION PROGRAM FOR SCALD SYSTEM

```
WAVE  0    10000
HISTORY    30000
OPEN       RD
OPEN       WR
OPEN       IORQ
OPEN       MEMRQ
OPEN       WAIT
OPEN       SOMEONE
OPEN       SF
OPEN       SET
OPEN       DEN
OPEN       DOEN
OPEN       A<2..0>;
OPEN       D<7..0>;
OPEN       DO<7..0>;
OPEN       CLK !C 0-1
LOGIC_INIT 0
OPEN       WR
DEPOSIT    1
OPEN       RD
DEPOSIT    1
OPEN       MEMRQ
DEPOSIT    1
OPEN       IORQ
DEPOSIT    1

     /* FILL MASK REGISTER WITH FF */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    0
SIMULATE   100
OPEN       D<7..0>;
DEPOSIT    FF
SIMULATE   100
OPEN       IORQ
DEPOSIT    0
OPEN       WR
DEPOSIT    0
SIMULATE   290
OPEN       IORQ
DEPOSIT    1
OPEN       WR
DEPOSIT    1
SIMULATE   150

     /* FILL COMPARAND REGISTER WITH 11 */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    1
SIMULATE   100
OPEN       D<7..0>;
DEPOSIT    11
SIMULATE   100
OPEN       IORQ
DEPOSIT    0
OPEN       WR
DEPOSIT    0
SIMULATE   290
OPEN       IORQ
DEPOSIT    1
```

```
OPEN       WR
DEPOSIT    1
SIMULATE   150

     /* WRITE 11 INTO CAM WORD 0 */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    0
SIMULATE   65
OPEN       MEMRQ
DEPOSIT    0
SIMULATE   175
OPEN       WR
DEPOSIT    0
SIMULATE   220
OPEN       WR
DEPOSIT    1
SIMULATE   20
OPEN       MEMRQ
DEPOSIT    1
SIMULATE   40

     /* FILL COMPARAND REGISTER WITH 33 */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    1
SIMULATE   100
OPEN       D<7..0>;
DEPOSIT    33
SIMULATE   100
OPEN       IORQ
DEPOSIT    0
OPEN       WR
DEPOSIT    0
SIMULATE   290
OPEN       IORQ
DEPOSIT    1
OPEN       WR
DEPOSIT    1
SIMULATE   150

     /* WRITE 33 INTO CAM WORD 1 */
SIMULATE   100
OPEN       A<2..0>;
DEPOSIT    1
SIMULATE   65
OPEN       MEMRQ
DEPOSIT    0
SIMULATE   175
OPEN       WR
DEPOSIT    0
SIMULATE   220
OPEN       WR
DEPOSIT    1
SIMULATE   20
OPEN       MEMRQ
DEPOSIT    1
SIMULATE   40

     /* FILL COMPARAND REGISTER WITH 77 */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    1
SIMULATE   100
OPEN       D<7..0>;
DEPOSIT    77
SIMULATE   100
OPEN       IORQ
DEPOSIT    0
```

```
OPEN      WR
DEPOSIT   0
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150

     /* WRITE 77 INTO CAM WORD 2 */
SIMULATE  100
OPEN      A<2..0>;
DEPOSIT   2
SIMULATE  65
OPEN      MEMRQ
DEPOSIT   0
SIMULATE  175
OPEN      WR
DEPOSIT   0
SIMULATE  220
OPEN      WR
DEPOSIT   1
SIMULATE  20
OPEN      MEMRQ
DEPOSIT   1
SIMULATE  40

     /* FILL COMPARAND REGISTER WITH FF */
SIMULATE  110
OPEN      A<2..0>;
DEPOSIT   1
SIMULATE  100
OPEN      D<7..0>;
DEPOSIT   FF
SIMULATE  100
OPEN      IORQ
DEPOSIT   0
OPEN      WR
DEPOSIT   0
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150

     /* WRITE FF INTO CAM WORD 3 */
SIMULATE  100
OPEN      A<2..0>;
DEPOSIT   3
SIMULATE  65
OPEN      MEMRQ
DEPOSIT   0
SIMULATE  175
OPEN      WR
DEPOSIT   0
SIMULATE  220
OPEN      WR
DEPOSIT   1
SIMULATE  20
OPEN      MEMRQ
DEPOSIT   1
SIMULATE  40

     /* SET SIMULATION */
SIMULATE  200
OPEN      SET
DEPOSIT   1
SIMULATE  350
```

```
OPEN       SET
DEPOSIT    0
SIMULATE   200
     /* FILL COMPARAND REGISTER WITH 11 */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    1
SIMULATE   100
OPEN       D<7..0>;
DEPOSIT    11
SIMULATE   100
OPEN       IORQ
DEPOSIT    0
OPEN       WR
DEPOSIT    0
SIMULATE   290
OPEN       IORQ
DEPOSIT    1
OPEN       WR
DEPOSIT    1
SIMULATE   150

     /READ 11 FROM CAM */
SIMULATE   185
OPEN       MEMRQ
DEPOSIT    0
SIMULATE   35
OPEN       RD
DEPOSIT    0
SIMULATE   905
OPEN       RD
DEPOSIT    1
SIMULATE   10
OPEN       MEMRQ
DEPOSIT    1
SIMULATE   115

     /* SET SIMULATION */
SIMULATE   200
OPEN       SET
DEPOSIT    1
SIMULATE   350
OPEN       SET
DEPOSIT    0
SIMULATE   200
     /* FILL COMPARAND REGISTER WITH 33 */
SIMULATE   110
OPEN       A<2..0>;
DEPOSIT    1
SIMULATE   100
OPEN       D<7..0>;
DEPOSIT    33
SIMULATE   100
OPEN       IORQ
DEPOSIT    0
OPEN       WR
DEPOSIT    0
SIMULATE   290
OPEN       IORQ
DEPOSIT    1
OPEN       WR
DEPOSIT    1
SIMULATE   150

     /READ 77 FROM CAM */
SIMULATE   185
OPEN       MEMRQ
DEPOSIT    0
SIMULATE   35
```

```
OPEN      RD
DEPOSIT   0
SIMULATE  905
OPEN      RD
DEPOSIT   1
SIMULATE  10
OPEN      MEMRQ
DEPOSIT   1
SIMULATE  115

     /* SET SIMULATION */
SIMULATE  200
OPEN      SET
DEPOSIT   1
SIMULATE  350
OPEN      SET
DEPOSIT   0
SIMULATE  200
     /* FILL COMPARAND REGISTER WITH FF */
SIMULATE  110
OPEN      A<2..0>;
DEPOSIT   1
SIMULATE  100
OPEN      D<7..0>;
DEPOSIT   FF
SIMULATE  100
OPEN      IORQ
DEPOSIT   0
OPEN      WR
DEPOSIT   0
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150

     /READ 33 FROM CAM */
SIMULATE  185
OPEN      MEMRQ
DEPOSIT   0
SIMULATE  35
OPEN      RD
DEPOSIT   0
SIMULATE  905
OPEN      RD
DEPOSIT   1
SIMULATE  10
OPEN      MEMRQ
DEPOSIT   1
SIMULATE  115

     /* SET SIMULATION */
SIMULATE  200
OPEN      SET
DEPOSIT   1
SIMULATE  350
OPEN      SET
DEPOSIT   0
SIMULATE  200
     /* FILL MASK REGISTER WITH 01 */
SIMULATE  110
OPEN      A<2..0>;
DEPOSIT   0
SIMULATE  100
OPEN      D<7..0>;
DEPOSIT   01
SIMULATE  100
OPEN      IORQ
```

```
DEPOSIT   0
OPEN      WR
DEPOSIT   0
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150

     /* FILL COMPARAND REGISTER WITH 01 */
SIMULATE  110
OPEN      A<2..0>;
DEPOSIT   0
SIMULATE  100
OPEN      D<7..0>;
DEPOSIT   01
SIMULATE  100
OPEN      IORQ
DEPOSIT   0
OPEN      WR
DEPOSIT   0
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150

     /READ 01 FROM CAM */
SIMULATE  185
OPEN      MEMRQ
DEPOSIT   0
SIMULATE  35
OPEN      RD
DEPOSIT   0
SIMULATE  905
OPEN      RD
DEPOSIT   1
SIMULATE  10
OPEN      MEMRQ
DEPOSIT   1
SIMULATE  115

     /* SET SIMULATION */
SIMULATE  200
OPEN      SET
DEPOSIT   1
SIMULATE  350
OPEN      SET
DEPOSIT   0
SIMULATE  200
     /* FILL MASK REGISTER WITH 30 */
SIMULATE  110
OPEN      A<2..0>;
DEPOSIT   0
SIMULATE  100
OPEN      D<7..0>;
DEPOSIT   30
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
```

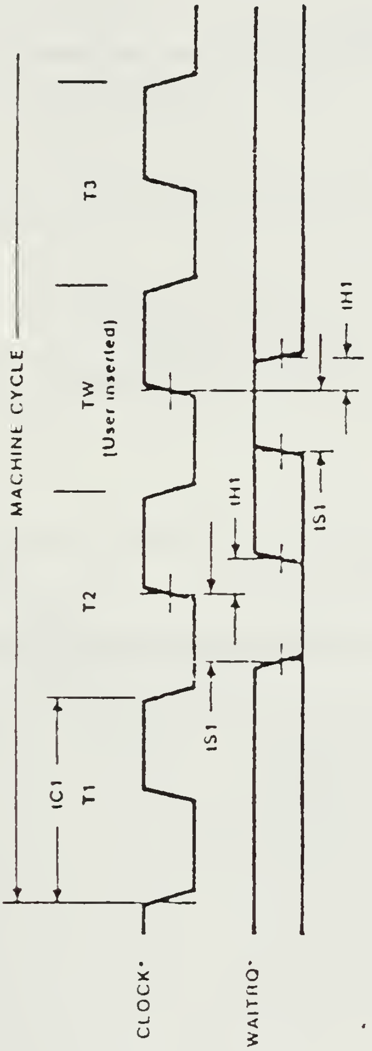
```
SIMULATE  150

     /* FILL COMPARAND REGISTER WITH 30 */
SIMULATE  110
OPEN      A<2..0>;
DEPOSIT   1
SIMULATE  100
OPEN      D<7..0>;
DEPOSIT   30
SIMULATE  100
OPEN      IORQ
DEPOSIT   0
OPEN      WR
DEPOSIT   0
SIMULATE  290
OPEN      IORQ
DEPOSIT   1
OPEN      WR
DEPOSIT   1
SIMULATE  150

     /READ 30 FROM CAM */
SIMULATE  185
OPEN      MEMRQ
DEPOSIT   0
SIMULATE  35
OPEN      RD
DEPOSIT   0
SIMULATE  905
OPEN      RD
DEPOSIT   1
SIMULATE  10
OPEN      MEMRQ
DEPOSIT   1
SIMULATE  115
```

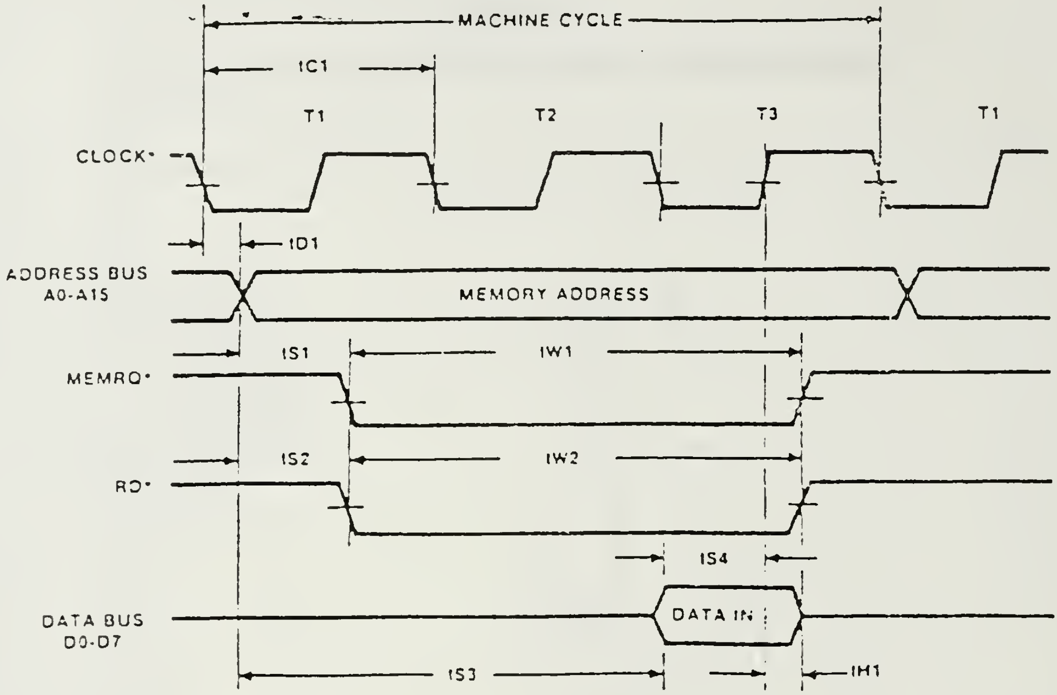
APPENDIX C

MICROPROCESSOR(Z-80) TIMING DIAGRAM



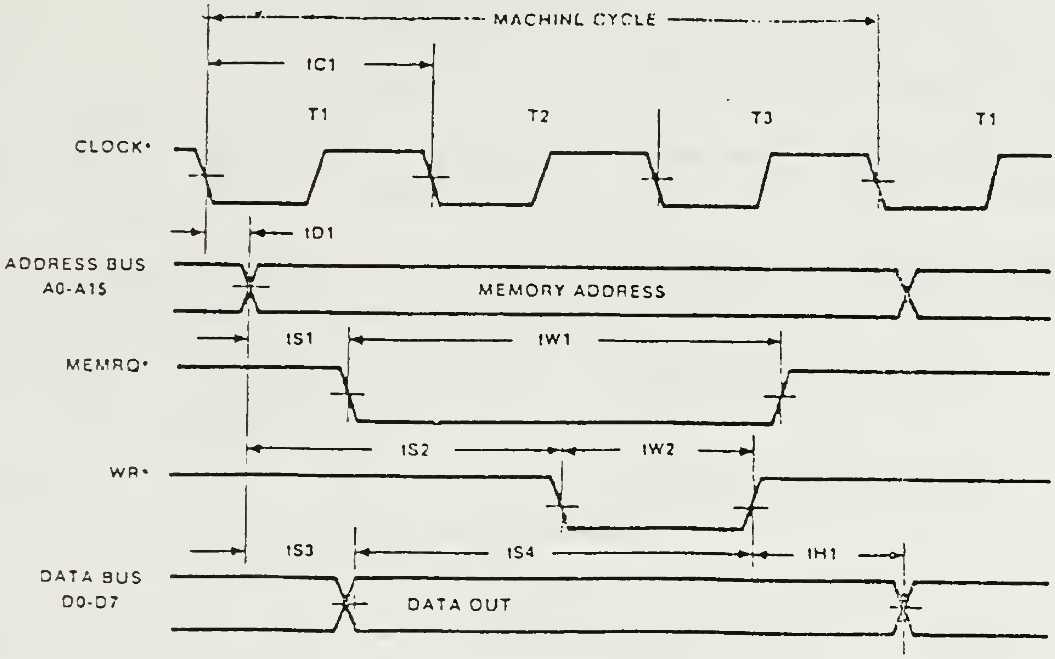
SYMBOL	PARAMETER	4.0 MHz	
		MIN	MAX
tC1	Clock cycle time	250	—
tH1	WAITRQ* hold after clock high	0	—
tS1	WAITRQ* setup before clock high	90	—

Figure C.1 WAIT Request Timing Diagram.



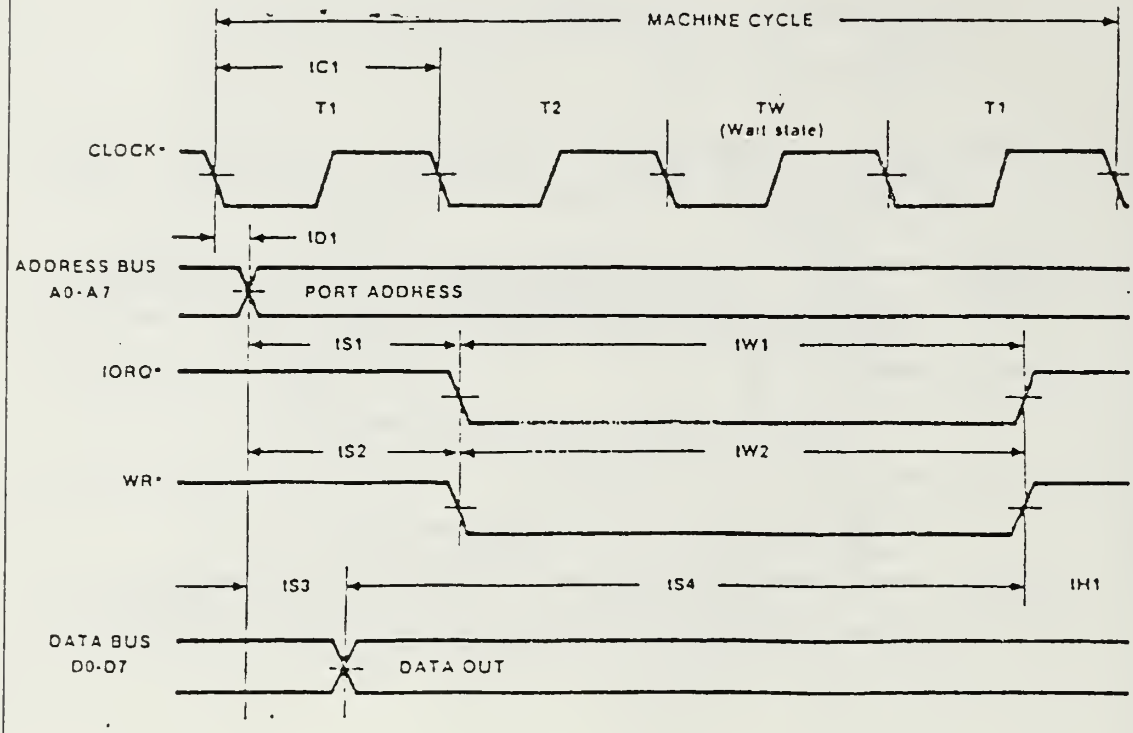
SYMBOL	PARAMETER	4.0 MHz	
		MIN	MAX
tC1	Clock cycle time	250	—
tD1	Address delay from T1 clock	—	110
tH1	Data hold time after clock high	0	—
tS1	Address setup before MEMRQ* active	75	—
tS2	Address setup before RD* active	110	—
tS3	Address setup before data valid	430	—
tS4	Data setup time before clock high	90	—
tW1	MEMRQ* pulse width	415	—
tW2	RD* pulse width	405	—

Figure C.2 Memory Read Cycle Timing Diagram.



SYMBOL	PARAMETER	4.0 MHz	
		MIN	MAX
tC1	Clock cycle time	250	—
tD1	Address delay from T1 clock	—	110
tH1	Data hold time after WR* ends	60	—
tS1	Address setup before MEMRQ* active	65	—
tS2	Address setup before WR* active	240	—
tS3	Address setup before data valid	175	—
tS4	Data setup time before WR* ends	300	—
tW1	MEMRQ* pulse width	415	—
tW2	WR* pulse width	220	—

Figure C.3 Memory Write Cycle Timing Diagram.



SYMBOL	PARAMETER	4.0 MHz	
		MIN	MAX
tC1	Clock cycle time	250	—
tD1	Address delay from T1 clock	—	110
tH1	Data hold time after WR* ends	0	—
tS1	Address setup before IORQ* active	180	—
tS2	Address setup before WR* active	205	—
tS3	Address setup before data valid	700	—
tS4	Data setup time before WR* ends	735	—
tW1	IORQ* pulse width	540	—
tW2	WR* pulse width	545	—

Figure C.4 Output Timing Diagram.

LIST OF REFERENCES

1. Pohm, A. V. and Agrawal, O. P., *High-Speed Memory Systems*, Reston Publishing, 1983.

2. Foster, C. Caxton, *Content Addressable Memory*, Van Nostrand Reinhold Company, 1976.

3. Tenenbaum, M. Aaron and others, *Data Structure for Personal Computer*, Prentice-Hall, 1985.

4. *PRO-LOG System User's Manual*, PRO-LOG System Inc., 1981.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
2.	Defence Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
3.	Department Chairman, Code 62 Dept. of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Prof. Chin-Hwa Lee, Code 62Le Dept. of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
5.	Prof. Sherif Micheal, Code 62Mi Dept. of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6.	Ham, Byung Woon Dept. of Electrical Engineering Naval Academy, Jinhae City, Gyungnam 602-00 Republic of Korea	8
7.	Naval Academy Library Jinhae City, Gyungnam 602-00 Republic of Korea	2

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

Thesis
H162175 Ham
c.1 Design of an addressable memory controller.

Thesis
H162175 Ham
c.1 Design of an addressable memory controller.

thesH162175
Design of an addressable memory controll
3 2768 000 72425 6
DUDLEY KNOX LIBRARY